



Министерство
образования
и науки РФ



Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук



Российский фонд
фундаментальных
исследований



Тверской
государственный технический
университет



ООО
«Технологии
системного анализа»

ТРУДЫ

IV Всероссийской научной конференции
молодых ученых с международным участием
“ИНФОРМАТИКА, УПРАВЛЕНИЕ И СИСТЕМНЫЙ АНАЛИЗ”



ТОМ I

Тверь
8-11 июня 2016 года

Министерство образования и науки Российской Федерации
Федеральный исследовательский центр «Информатика и управление»
Российской академии наук
Тверской государственной технической университет
ООО «Технологии системного анализа»

IV Всероссийская научная
конференция молодых ученых
с международным участием

ИНФОРМАТИКА, УПРАВЛЕНИЕ И СИСТЕМНЫЙ АНАЛИЗ

Труды IV Всероссийской научной конференции
молодых ученых с международным участием

ТОМ I

8 – 11 июня 2016 г.

Тверь

Информатика, управление и системный анализ: Труды IV Всероссийской научной конференции молодых ученых с международным участием. – Т. I. – Тверь: Тверской государственной технический университет, 2016. – 224 с.

В сборнике трудов конференции представлены результаты исследований молодых ученых ведущих научных организаций России по широкому спектру актуальных проблем информатики, управления и системного анализа. Междисциплинарные исследования и интенсивное использование данных - ключевые особенности современной науки. Конференция ИУСА-2016 ставит своей целью наладить устойчивые связи между молодыми учеными, работающими в совершенно разных научных областях, но использующих при этом сходные методы и инструменты. Налаживание такого сетевого взаимодействия может обеспечить существенные прорывы в исследованиях за счет обмена опытом с молодыми и старшими коллегами в рамках мероприятий, проводимых на конференции.

Конференция ИУСА-2016 продолжает традиции молодежных конференций «Теория и практика системного анализа» (ТПСА), которые проводились в период с 2010 по 2014 годы.

В организации конференции принимали участие:

- Федеральный исследовательский центр «Информатика и управление» Российской академии наук;
- Тверской государственной технический университет;
- ООО «Технологии системного анализа».

Конференция поддержана: Российским фондом фундаментальных исследований (грант № 16-37-10178), Российской академией наук, Федеральным агентством научных организаций.

ПРОГРАММНЫЙ КОМИТЕТ

Председатель:

- Соколов И.А., академик РАН (ФИЦ ИУ РАН, Россия)

Сопредседатели:

- Попков Ю.С., чл.-корр. РАН (ФИЦ ИУ РАН, Россия)
- Твардовский А. В. д.ф.-м.н. (ТвГТУ, Россия)

Заместитель председателя:

- Осипов Г.С., д.ф.-м.н. (ФИЦ ИУ РАН, Россия)

Члены комитета:

- Сгурев В., академик (Болгарская академия наук, Болгария)
- Арлазаров В.Л., чл.-корр. РАН (ФИЦ ИУ РАН, Россия)
- Новиков Д.А., чл.-корр. РАН (ИПУ РАН, Россия)
- Ильин А.В., чл.-корр. РАН (МГУ, Россия)
- Палюх Б.В., д.т.н. (ТвГТУ, Россия)
- Железны М., PhD (Западнечешский Университет, Чехия)
- Филаткина Н., PhD (Трирский Университет, Германия)
- Мельников А.В., д.т.н. (ЧелГУ, Россия)
- Лившиц В.Н., д.э.н. (ФИЦ ИУ РАН, Россия)
- Посыпкин М. А., д.ф.-м.н. (ФИЦ ИУ РАН, Россия)
- Матвеев И. А., д.т.н. (ФИЦ ИУ РАН, Россия)
- Борисов А. В., д.ф.-м.н. (ФИЦ ИУ РАН, МАИ, Россия)
- Вохминцев А.В., к.т.н. (ЧелГУ, Россия)
- Бритков В.Б., к.ф.-м.н. (ФИЦ ИУ РАН, Россия)
- Тихомиров И.А., к.т.н. (ФИЦ ИУ РАН, Россия)
- Смирнов И.В., к.ф.-м.н. (ФИЦ ИУ РАН, Россия)
- Трусова Ю.О., к.т.н. (ФИЦ ИУ РАН, Россия)
- Горшенин А.К., к.ф.-м.н. (ФИЦ ИУ РАН, Россия)
- Булычев А.В., к.т.н. (ФИЦ ИУ РАН, Россия)
- Ройзензон Г.В., к.т.н. (ФИЦ ИУ РАН, Россия)
- Евстигнеев Н.М., к.ф.-м.н. (ФИЦ ИУ РАН, Россия)
- Курочкин И.И., к.т.н. (ИППИ РАН, Россия)

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

Председатель:

- Тихомиров И. А. (ФИЦ ИУ РАН)

Заместитель председателя:

- Рассадин С. В. (ТвГТУ)
- Булычев А. В. (ФИЦ ИУ РАН)
- Иванников А. Ф. , к.ф.-м.н. (ТвГТУ)

Члены Оргкомитета:

- Жарова Н. Ю. (ФИЦ ИУ РАН),
- Трусова Ю. О. (ФИЦ ИУ РАН)
- Горшенин А. К. (ФИЦ ИУ РАН)
- Яковлев К. С. (ФИЦ ИУ РАН)
- Тарханов И. А. (ФИЦ ИУ РАН)
- Панов А. И. (ФИЦ ИУ РАН)
- Святкина М. Н. (МГТУ им. Н.Э. Баумана)
- Ананьева М. И. (ООО «ТСА»)
- Парадовская Л. И. (ООО «ТСА»)
- Зыков И. И. (ТвГТУ)
- Ветров А. Н. (ТвГТУ)
- Лебедев И. С. (ТвГТУ)
- Тарасов В. Б. (МГТУ им. Н.Э. Баумана)

СЕКЦИЯ «АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ»

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ БИЛИНЕЙНОЙ ДИСКРЕТНОЙ СТОХАСТИЧЕСКОЙ СИСТЕМОЙ ПО ВЕРОЯТНОСТНОМУ ИНТЕГРАЛЬНОМУ КРИТЕРИЮ

В.М. Азанов

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский авиационный институт (национальный исследовательский университет)» (МАИ)
107014, г. Москва, ул. Малая Остроумовская, д. 1г., кв. 44. E-mail: azanov59@gmail.com

Ю.С. Кан

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский авиационный институт (национальный исследовательский университет)» (МАИ)
143500, Московская обл., г. Истра, ул. Юбилейная, д. 7, кв. 62. E-mail: yu_kan@mail.ru

Аннотация

Рассматривается задача синтеза оптимального управления дискретной стохастической билинейной системой. Случайные возмущения в системе играют роль ошибок реализации скалярного управляющего воздействия. Критерием оптимальности служит вероятность пребывания линейной комбинации вектора состояния в заданной окрестности нуля в течении фиксированного времени. С использованием метода динамического программирования разрабатывается вспомогательный аппарат для решения задачи. Для случая некоторого класса систем приводится ее аналитическое решение. В качестве примера рассматривается задача оптимизации маневра искусственного спутника Земли в окрестности круговой орбиты.

1. Введение

Задачи оптимального управления при наличии случайных возмущений представляют практический интерес в связи с необходимостью формирования алгоритмов управления движущимися объектами. Важным при решении задачи синтеза оптимального управления является учет неконтролируемых факторов, возникающих, в частности, при обработке управляющих сигналов (см., например, [1-3]).

Если априори известен лишь диапазон изменения неконтролируемых ошибок, применяют минимаксную постановку, при которой показателем качества является наихудшее по неопределенным факторам значение целевого функционала (поэтому такая постановка также называется гарантирующей). Иногда [4] используется аналог "вероятностно-гарантирующей" постановки, а именно, неконтролируемые факторы предполагаются случайными, имеющими равномерный закон распределения. Основанием для рассмотрения равномерного закона распределения выступает принцип равномерности [5], [6], который гласит, что плотность равномерного распределения оказывается наихудшей с точки зрения вероятностного показателя на некотором естественном классе плотностей. При этом в качестве критерия качества рассматривается функционал вероятности, или функционал квантили [1].

Функционал вероятности представляет собой вероятность неперевышения некоторым точностным функционалом заданного допустимого уровня. Сам точностной функционал при этом характеризует точность системы управления.

В настоящей работе рассматривается задача оптимального управления дискретной стохастической системой с критерием, представляющим собой вероятность пребывания линейной комбинации вектора состояния в окрестности нуля в течении заданного горизонта управления. Случайные возмущения играют роль ошибок исполнения управляющих воздействий. На основе метода динамического программирования (МДП) разрабатывается вспомогательный аппарат для решения задачи. Для случая некоторого класса систем приводится аналитическое решение задачи, которое сравнивается с известным решением на примере задачи удержания искусственного спутника Земли в окрестности круговой орбиты. Последняя представляет практический и теоретический интерес в настоящее время [7-9].

2. Постановка задачи

Рассмотрим стационарную дискретную стохастическую систему

$$\begin{cases} x_{k+1} = Ax_k + b(u_k(1 + \xi_k)), & k = \overline{0, N}, \\ x_0 = X, \end{cases} \quad (1)$$

где $x_k = (x_k^1, \dots, x_k^n) \in R^n$ - вектор состояния, X - известный вектор; $u_k \in Q \subset R$ - скалярное управление, на которое наложены геометрические ограничения $|u_k| \leq q$, $q > 0$ - известный скаляр; $\xi_k \in R$ - независимые случайные величины (с.в.), моделирующие относительную неточность реализации управляющего воздействия. В силу мультипликативности случайных помех к управлению, систему (1) также называют билинейной, т.е. линейной по вектору состояния, по управлению при фиксированных случайных помехах и по случайным помехам при фиксированном управлении. Предполагается, что носители плотностей распределения с.в. ξ_k являются отрезками, $\text{supp} f_k^{\xi} = [-\varepsilon_k, \varepsilon_k]$, $\varepsilon_k < 1$. Функции распределения ξ_k будем обозначать F_k^{ξ} . $A \in R^{n \times n}$, $b \in R^n$ - известные матрицы, $\det A \neq 0$. Предполагается, что пара (A, b) управляема.

На траекториях системы (1) определим функционал

$$P_{\varphi}(u(\cdot)) = \mathbf{P} \left(\max_{k \in \{0, \dots, N\}} |l^T x_{k+1}| \leq \varphi \right),$$

Где $u(\cdot) = (u_0(\cdot), \dots, u_N(\cdot))$ - управление, которое ищется в классе функций текущего состояния $u_k = u_k(x_k)$; \mathbf{P} - вероятность; $\varphi > 0$ - фиксированный скаляр, $l \in R^n$ - детерминированный вектор.

Функционал $P_{\varphi}(u(\cdot))$ характеризует вероятность того, что модуль линейной комбинации координат вектора состояния не превысит заданное значение φ в каждый момент времени на заданном горизонте N . Использование линейной комбинации в функционале вероятности позволяет в удобной форме учесть какой именно параметр системы требуется удержать в заданном диапазоне.

Ставится задача

$$P_{\varphi}(u(\cdot)) \rightarrow \max_{u(\cdot) \in U} \quad (2)$$

где U - область допустимых управлений

$$U = \underbrace{Q \times \dots \times Q}_{N+1}, \quad Q = \{v \in R: |v| \leq q\}.$$

Задачу (2) можно интерпретировать, как задачу удержания системы (1) на множестве

$$B_{N+1}^\varphi = \{x \in R^n : |l^T x| \leq \varphi\}.$$

Далее будет приведен вспомогательный аппарат для решения задачи (2), основанный на методе динамического программирования (МДП) и его некоторой модификации.

3. Уравнения метода динамического программирования и их аналог для поверхностей уровня 1 функции Беллмана

Для решения задачи (2) воспользуемся МДП. Важно отметить, что в [1] сформулированы достаточные условия существования равномерно оптимальной марковской стратегии для задач с вероятностным терминальным критерием. Настоящую задачу можно преобразовать к указанной путем введения новой переменной

$$y_k = \max_{i \in \{1, \dots, k\}} |l^T x_i|, \quad y_0 = 0,$$

и расширения вектора состояния

$$z_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}, \quad f_k(z_k, u_k, \xi_k) = \begin{pmatrix} Ax_k + b(u_k(1 + \xi_k)) \\ \max\{y_k, |l^T x_{k+1}|\} \end{pmatrix},$$

$$z_{k+1} = f_k(z_k, u_k, \xi_k), \quad k = \overline{0, N}.$$

При этом эквивалентная задача в пространстве более высокой размерности запишется в виде

$$\mathbf{P}(y_{N+1} \leq \varphi) \rightarrow \max_{u(\cdot) \in U}.$$

Поскольку $f_k(z_k, u_k, \xi_k)$ - непрерывна, множество Q - компакт и выполнено

$$\mathbf{P}(y_{N+1} \geq 0) = 1,$$

решение задачи (2) существует в классе марковских стратегий и вычисляется в результате решения следующих конечномерных задач

$$W_k^\varphi(z_k) = \max_{u_k \in Q} \mathbf{M}[W_{k+1}^\varphi(f_k(z_k, u_k, \xi_k))], \quad k = \overline{0, N},$$

$$W_{N+1}^\varphi(z_{N+1}) = \mathbf{I}_{[0, \varphi]}(y_{N+1}),$$

где $W_k^\varphi(z_k)$ - функция Беллмана в приведенной задаче, $\mathbf{I}_A(z)$ - индикаторная функция множества A , $\mathbf{M}[\cdot]$ - оператор математического ожидания.

Однако можно обойтись без указанного приема, если определить функцию Беллмана следующим образом

$$B_k^\varphi(x) = \sup_{u_k(\cdot) \in Q, \dots, u_N(\cdot) \in Q} \mathbf{P}\left(\max_{i \in \{k, \dots, N\}} |l^T x_{i+1}(x_k, u_k(\cdot), \dots, u_i(\cdot), \xi_k, \dots, \xi_i)| \leq \varphi \mid x_k = x\right).$$

Уравнения МДП в данном случае удается преобразовать к следующему виду

$$B_k^\varphi(x_k) = \max_{u_k \in Q} \mathbf{M}\left[\mathbf{I}_{[0, \varphi]}(|l^T x_k|) B_{k+1}^\varphi(f_k(x_k, u_k, \xi_k))\right], \quad k = \overline{1, N},$$

$$B_0^\varphi(x_0) = \max_{u_0 \in Q} \mathbf{M}\left[B_1^\varphi(f_0(x_0, u_0, \xi_0))\right], \quad (3)$$

$$B_{N+1}^\varphi(x_{N+1}) = \mathbf{I}_{[0, \varphi]}(|l^T x_{N+1}|).$$

Введем в рассмотрение поверхности уровней 1 и 0 функции Беллмана

$$\mathbf{B}_k^\varphi = \{x \in R^n : B_k^\varphi(x) = 1\}, \quad \mathbf{O}_k^\varphi = \{x \in R^n : B_k^\varphi(x) = 0\}, \quad k = \overline{0, N}.$$

Нетрудно видеть, что

$$\mathbf{B}_{N+1}^\varphi = \{x \in R^n : |l^T x| \leq \varphi\}, \quad \mathbf{O}_{N+1}^\varphi = \{x \in R^n : |l^T x| > \varphi\}.$$

Из формулы (3) также следует, что $\forall k = \overline{0, N}$ выполнено $\mathbf{O}_{N+1}^\varphi \subset \mathbf{O}_k^\varphi$.

Утверждение 1. Множества \mathbf{B}_k^φ непусты и вычисляются в соответствии с рекуррентными соотношениями в дискретном времени $k = \overline{0, N}$

$$\begin{aligned} \mathbf{B}_k^\varphi &= \{x \in R^n : \exists u = u(x) \in Q : \mathbf{P}(f_k(x, u, \xi_k) \in \mathbf{B}_{k+1}^\varphi) = 1\} \cap \mathbf{B}_{N+1}^\varphi, \\ \mathbf{B}_{N+1}^\varphi &= \{x \in R^n : |l^T x| \leq \varphi\}. \end{aligned} \quad (4)$$

Доказательство. Непустоту множеств легко проверить, положив $u_k = 0$ для всех $k = \overline{0, N}$, при этом функция Беллмана примет вид

$$B_k^\varphi(x_k) = \prod_{i=k}^{N+1} \mathbf{I}_{[0, \varphi]}(|l^T A^{N+1-i} x_k|) = \mathbf{I}_{[0, \varphi]} \left(\max_{i \in \{k, \dots, N+1\}} |l^T A^{N+1-i} x_k| \right),$$

и следовательно

$$\{x \in R^n : \max_{i \in \{k, \dots, N+1\}} |l^T A^{N+1-i} x| \leq \varphi\} \subset \mathbf{B}_k^\varphi.$$

Из определения поверхности уровня 1 функции Беллмана следует

$$\begin{aligned} \mathbf{B}_k^\varphi &= \left\{ x \in R^n : \max_{u_k \in Q} \mathbf{M} \left[\mathbf{I}_{\mathbf{B}_{N+1}^\varphi}(x) B_{k+1}^\varphi(f_k(x, u_k, \xi_k)) \right] = 1 \right\} = \\ &= \left\{ x \in R^n : \max_{u_k \in Q} \mathbf{M} \left[B_{k+1}^\varphi(f_k(x, u_k, \xi_k)) \right] = 1 \right\} \cap \{x \in R^n : |l^T x| \leq \varphi\}. \end{aligned}$$

Из непустоты множеств \mathbf{B}_k^φ следует, что функцию Беллмана на некотором шаге $k+1$ можно представить в следующем виде

$$B_{k+1}^\varphi(x_{k+1}) = \mathbf{I}_{\mathbf{B}_{N+1}^\varphi}(x_{k+1}) \left(\mathbf{I}_{\mathbf{B}_{k+1}^\varphi}(x_{k+1}) + (1 - \mathbf{I}_{\mathbf{B}_{k+1}^\varphi}(x_{k+1})) V_{k+1}^\varphi(x_{k+1}) \right), \quad (5)$$

где $V_{k+1}^\varphi : R^n \rightarrow R$ - некоторая функция, сужение которой на множестве R^n , \mathbf{B}_{k+1}^φ принимает значение в диапазоне $[0, 1)$. Переходя к следующему шагу МДП, используя формулу полного математического ожидания и включение $\mathbf{B}_{N+1}^\varphi \subset \mathbf{B}_{k+1}^\varphi$, имеем

$$\begin{aligned} \max_{u_k \in Q} \mathbf{M} \left[B_{k+1}^\varphi(f_k(x_k, u_k, \xi_k)) \right] &= \max_{u_k \in Q} \mathbf{I}_{\mathbf{B}_{N+1}^\varphi}(x_k) \left(\mathbf{P}(f_k(x_k, u_k, \xi_k) \in \mathbf{B}_{k+1}^\varphi) + \right. \\ &\left. + (1 - \mathbf{P}(f_k(x_k, u_k, \xi_k) \in \mathbf{B}_{k+1}^\varphi)) \mathbf{M} \left[V_{k+1}^\varphi(f_k(x_k, u_k, \xi_k)) \mid \{f_k(x_k, u_k, \xi_k) \notin \mathbf{B}_{k+1}^\varphi\} \right] \right). \end{aligned}$$

Поскольку для условного математического ожидания в вышезаписанной формуле выполнено

$$\mathbf{M} \left[V_{k+1}^\varphi(f_k(x_k, u_k, \xi_k)) \mid \{f_k(x_k, u_k, \xi_k) \notin \mathbf{B}_{k+1}^\varphi\} \right] \in [0, 1),$$

функция Беллмана принимает значение 1 только для таких x_k, u_k , для которых одновременно выполнено

$$\mathbf{I}_{\mathbf{B}_{N+1}^\varphi}(x_k) = 1, \quad \mathbf{P}(f_k(x_k, u_k, \xi_k) \in \mathbf{B}_{k+1}^\varphi) = 1,$$

причем такие управления u_k являются оптимальными для $x_k \in \mathbf{B}_k^\varphi$.

Утверждение доказано.

Интересно, что из доказательства утверждения можно модифицировать уравнения МДП, принимая во внимание представление функции Беллмана (5). По сути это представление позволяет рассматривать функцию Беллмана только в области $x_k \in \square^n$, B_k^φ . Имеем

$$\begin{aligned} B_k^\varphi(x_k) &= \max_{u_k \in Q} \mathbf{I}_{B_{N+1}^\varphi}(x_k) \left(\mathbf{I}_{B_k^\varphi}(x_k) + \left(1 - \mathbf{I}_{B_k^\varphi}(x_k)\right) \mathbf{P}(f_k(x_k, u_k, \xi_k) \in B_{k+1}^\varphi) \right) + \\ &+ \left(1 - \mathbf{I}_{B_k^\varphi}(x_k)\right) \left(1 - \mathbf{P}(f_k(x_k, u_k, \xi_k) \in B_{k+1}^\varphi)\right) \mathbf{M} \left[V_{k+1}^\varphi(f_k(x_k, u_k, \xi_k)) \middle| \{f_k(x_k, u_k, \xi_k) \notin B_{k+1}^\varphi\} \right] = \\ &= \mathbf{I}_{B_{N+1}^\varphi \cap B_k^\varphi}(x_k) + \max_{u_k \in Q} \mathbf{I}_{B_{N+1}^\varphi}(x_k) \left(1 - \mathbf{I}_{B_k^\varphi}(x_k)\right) \left(\mathbf{P}(f_k(x_k, u_k, \xi_k) \in B_{k+1}^\varphi) \right) + \\ &+ \left(1 - \mathbf{P}(f_k(x_k, u_k, \xi_k) \in B_{k+1}^\varphi)\right) \mathbf{M} \left[V_{k+1}^\varphi(f_k(x_k, u_k, \xi_k)) \middle| \{f_k(x_k, u_k, \xi_k) \notin B_{k+1}^\varphi\} \right]. \end{aligned}$$

Из последнего выражения видно, что если

$$\mathbf{I}_{B_{N+1}^\varphi}(x_k) \left(1 - \mathbf{I}_{B_k^\varphi}(x_k)\right) = \mathbf{I}_{B_{N+1}^\varphi}(x_k) \mathbf{I}_{R^n \setminus B_k^\varphi}(x_k) = 0,$$

или, что то же самое

$$B_{N+1}^\varphi \cap \{R^n \setminus B_k^\varphi\} = \emptyset, \quad (6)$$

то функция Беллмана равна

$$B_k^\varphi(x_k) = \mathbf{I}_{B_{N+1}^\varphi \cap B_k^\varphi}(x_k) = \mathbf{I}_{B_k^\varphi}(x_k).$$

Условие (6) также представимо в следующем виде

$$B_k^\varphi \cup O_k^\varphi = R^n.$$

Важно отметить, что для вычисления множеств B_k^φ не нужно вычислять саму функцию Беллмана (последнее является очень трудной задачей), что следует из утверждения 1. Причем в настоящей задаче структура B_k^φ зависит только от матриц l , A , b , и чисел q , ε_k .

Таким образом, если для всех $k = \overline{0, N}$ выполнено (6), то соотношения МДП допускают еще одну модификацию

$$\begin{aligned} B_k^\varphi(x_k) &= \max_{u_k \in Q} \mathbf{P}(f_k(x_k, u_k, \xi_k) \in B_{k+1}^\varphi), \quad k = \overline{0, N}. \\ u_k^\varphi(x_k) &= \arg \max_{u_k \in Q} \mathbf{P}(f_k(x_k, u_k, \xi_k) \in B_{k+1}^\varphi), \end{aligned} \quad (7)$$

что полностью ликвидирует проблему вычисления кратных интегралов в правой части соотношений (3).

В следующем разделе находится явный вид множеств B_k^φ , выделяется класс систем, для которых справедливо условие (6) и для данного класса решается исходная задача в соответствии с (7).

4. Решение задачи для узкого класса систем

Из условий управляемости матриц системы (1) существует число $N_u \in \{0, \dots, N\}$, такое, что

$$\begin{cases} l^T A^{N-N_u} b \neq 0, \\ l^T A^{N-1-N_u} b = 0, \end{cases} \quad \text{при } N_u \in \{0, \dots, N-1\}, \\ l^T b \neq 0, \quad \text{при } N_u = N.$$

Посути N_u является последним моментом реакции функционала $P_\varphi(u(\cdot))$ на управление. Нетрудно видеть, что при $N_u \neq N$, $\forall k > N_u$ функция Беллмана имеет вид

$$B_k^\varphi(x_k) = \prod_{i=k}^{N+1} \mathbf{I}_{[0,\varphi]}(|l^T A^{N+1-i} x_k|) = \mathbf{I}_{[0,\varphi]} \left(\max_{i \in \{k, \dots, N+1\}} |l^T A^{N+1-i} x_k| \right),$$

и следовательно для $k = N_u + 1$ получаем

$$B_{N_u+1}^\varphi = \left\{ x \in R^n : \max_{i \in \{N_u+1, \dots, N+1\}} |l^T A^{N+1-i} x_k| \leq \varphi \right\}.$$

Перейдем к следующему шагу $k = N_u$. В соответствии с утверждением 1, получаем, что

$$B_{N_u}^\varphi = \left\{ x \in R^n : \exists u = u(x) \in Q : \mathbf{P} \left(|l^T A^{N+1-N_u} x + (l^T A^{N-N_u} b) u (1 + \xi_{N_u})| \leq \varphi \right) = 1 \right\} \cap B_{N_u+1}^\varphi.$$

В силу финитности плотностей вероятности с.в. ξ_k , на основе результатов работы [4] получаем, что уравнение

$$u \in Q : \mathbf{P} \left(|l^T A^{N+1-N_u} x_k + (l^T A^{N-N_u} b) u (1 + \xi_{N_u})| \leq \varphi \right) = 1$$

имеет бесконечное множество решений, лежащих на отрезке

$$U_{N_u} = \left\{ u \in Q : \max \left\{ 0, \frac{-\varphi + |l^T A^{N+1-N_u} x_{N_u}|}{|l^T A^{N-N_u} b| (1 - \varepsilon_{N_u})} \right\} \leq |u| \leq \min \left\{ q, \frac{\varphi + |l^T A^{N+1-N_u} x_{N_u}|}{|l^T A^{N-N_u} b| (1 + \varepsilon_{N_u})} \right\} \right\},$$

а поверхность уровня 1 функции Беллмана определяется выражением

$$B_{N_u}^\varphi = \left\{ x \in R^n : \max \left\{ |l^T A^{N+1-N_u} x| \varepsilon_{N_u}, |l^T A^{N+1-N_u} x| - q |l^T A^{N-N_u} b| (1 - \varepsilon_{N_u}) \right\} \leq \varphi \right\} \cap B_{N_u+1}^\varphi.$$

Отметим, что для «режима» управления $x_{N_u} \in B_{N_u}^\varphi$ оптимальным является любое управление из отрезка U_{N_u} . На примере будет показано, что при наличии некоторых дополнительных возмущений в системе выбором характерного управления из отрезка U_{N_u} можно существенно улучшить качество управления.

Для определения оптимального для $x_{N_u} \in B_{N_u+1}^\varphi \cap \{R^n, B_{N_u}^\varphi\}$ управления требуется конкретизировать распределение случайного возмущения ξ_{N_u} . Ниже приведено решение для случая равномерного распределения, являющегося важным случаем [4], [5], [6] в технических приложениях. Соответствующая задача стохастического программирования имеет вид

$$\mathbf{P} \left(|l^T A^{N+1-N_u} x_{N_u} + (l^T A^{N-N_u} b) u (1 + \xi_{N_u})| \leq \varphi \right) \rightarrow \max_{|u| \leq q}. \quad (8)$$

Утверждение 2. Пусть $\xi_{N_u} \sim R[-\varepsilon_{N_u}, \varepsilon_{N_u}]$, тогда для $x_{N_u} \in B_{N_u+1}^\varphi \cap \{R^n, B_{N_u}^\varphi\}$ оптимальным является управление

$$|u_{N_u}^\varphi| = \min \left\{ q, \frac{\varphi + |l^T A^{N+1-N_u} x_{N_u}|}{|l^T A^{N-N_u} b| (1 + \varepsilon_{N_u})} \right\}, \quad \text{sign}(u_{N_u}^\varphi) = -\text{sign} \left(\frac{l^T A^{N+1-N_u} x_{N_u}}{l^T A^{N-N_u} b} \right),$$

а функция Беллмана для $x_{N_u} \in B_{N_u+1}^\varphi \cap \{R^n, B_{N_u}^\varphi\}$ определяется выражением

$$V_{N_u}^\varphi(x_{N_u}) = \min \left\{ \frac{\varphi}{\varphi + |l^T A^{N+1-N_u} x_{N_u}|} \frac{1 + \varepsilon_{N_u}}{\varepsilon_{N_u}}, \max \left\{ 0, \frac{1}{2\varepsilon_{N_u}} \left(1 + \varepsilon_{N_u} + \frac{\varphi - |l^T A^{N+1-N_u} x_{N_u}|}{|l^T A^{N-N_u} b| q} \right) \right\} \right\}.$$

Доказательство утверждения 2 опирается на решение задачи стохастического программирования (8), которая без учета ограничений решена в [4].

С учетом полученного явного вида поверхности уровня 1 функции Беллмана для $k = N_u$ запишем условия, при которых выполнено (6):

$$\forall x: \max \left\{ |l^T A^{N+1-N_u} x| \varepsilon_{N_u}, \quad |l^T A^{N+1-N_u} x| - q |l^T A^{N-N_u} b| (1 - \varepsilon_{N_u}) \right\} \leq |l^T A^{N-N_u} x|.$$

Для выполнения указанных условий достаточно потребовать

$$(E - \varepsilon_{N_u} A^{-1}) l^T (E + \varepsilon_{N_u} A) \pm 0, \quad q \geq \varphi \left(|l^T A^{N-N_u} b| \varepsilon_{N_u} \right)^{-1}.$$

Если это справедливо для всех ε_k , то в соответствии с (7) и с учетом структуры поверхности уровня 1 функции Беллмана, оптимальное управление для всех k будет совпадать с управлением для $k = N_u$ с точностью до параметров ε_k

$$|u_k^\varphi| = \begin{cases} U_k, & x_k \in B_k^\varphi, \\ \min \left\{ q, \frac{\varphi + |l^T A^{N+1-N_u} x_k|}{|l^T A^{N-N_u} b| (1 + \varepsilon_k)} \right\}, & x_k \notin B_k^\varphi, \end{cases} \quad \text{sign}(u_k^\varphi) = -\text{sign} \left(\frac{l^T A^{N+1-N_u} x_k}{l^T A^{N-N_u} b} \right), \quad (9)$$

где

$$U_k = \left\{ u \in \mathcal{Q}: \max \left\{ 0, \frac{-\varphi + |l^T A^{N+1-N_u} x_k|}{|l^T A^{N-N_u} b| (1 - \varepsilon_k)} \right\} \leq |u| \leq \min \left\{ q, \frac{\varphi + |l^T A^{N+1-N_u} x_k|}{|l^T A^{N-N_u} b| (1 + \varepsilon_k)} \right\} \right\},$$

а $u_k^\varphi = U_k$ в (9) означает, что любое управление из указанного множества является оптимальным. При этом в случае $\varepsilon_k = \varepsilon$ получаем, что закон управления (9) не зависит явно от времени.

5. Оптимизация маневра искусственного спутника Земли в окрестности круговой орбиты

Рассмотрим модель движение ИСЗ в окрестности круговой орбиты [7], [8]

$$\begin{cases} x_{k+1}^1 = x_k^1 + \delta t x_k^2 + \delta t u_k (1 + \xi_k) + \eta_k^1, \\ x_{k+1}^2 = x_k^2 + u_k (1 + \xi_k) + \eta_k^2, \end{cases} \quad k = \overline{0, N},$$

где x_k^1 - отклонение гринвической долготы восходящего узла (ГДВУ) от требуемого значения (в градусах), x_k^2 - скорость дрейфа ГДВУ (в градусах/звездные сутки), $x_0 = X \sim N(m_X, \text{diag} \sigma_X)$ - гауссовский случайный вектор с известным математическим ожиданием и ковариационной матрицей, $\eta_k = (\eta_k^1, \eta_k^2)^T \sim N(0, \text{diag} \sigma_\eta)$ - гауссовский случайный вектор с нулевым средним и известной ковариационной матрицей, играющий роль ошибок измерения вектора состояния, $\delta t \in \square$ - интервал времени между коррекциями (в звездных сутках). Отличаем рассматриваемой системы от (1) является наличие аддитивного возмущения η_k и отсутствие ограничения на управляющее воздействие.

Целью управления является максимизировать вероятность пребывания ГДВУ в заданном диапазоне φ . Такую задачу можно трактовать как задачу удержания ИСЗ в окрестности круговой орбиты [7], [8]. В [7], [8] в качестве закона управления, решающего указанную выше задачу, использовалось следующее

$$u_k^M = -\frac{x_k^1 + \delta t x_k^2}{\delta t (1 + D[\xi_k])}$$

Функция u_k^M для $k = N$ является оптимальным управлением в задаче

$$\mathbf{M} \left[(x_{N+1}^1)^2 \right] \rightarrow \min_{u^{(\cdot)}}$$

т.е. в задаче со среднеквадратическим критерием. Важно отметить, что она найдена с учетом аддитивных случайных возмущений. Попытаемся выполнить цель управления с помощью управления, построенного без учета η_k в предыдущем разделе

В рамках исходных обозначений имеем

$$l = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} \delta t \\ 1 \end{pmatrix}, \quad N_u = N.$$

Рассмотрим два случая выбора характерных режимов управления на оптимальном отрезке U_k :

1) управление с зоной нечувствительности

$$|u_k^o| = \begin{cases} 0, & |l^T A x_k| \leq \varphi, \\ \min \left\{ \frac{-\varphi + |l^T A x_k|}{|l^T b|(1-\varepsilon)}, \frac{\varphi + |l^T A x_k|}{|l^T b|(1+\varepsilon)} \right\}, & |l^T A x_k| > \varphi, \end{cases} \quad \text{sign}(u_k^o) = -\text{sign} \left(\frac{l^T A x_k}{l^T b} \right).$$

2) управление с линейным участком

$$|u_k^L| = \min \left\{ \frac{|l^T A x_k|}{|l^T b|}, \frac{\varphi + |l^T A x_k|}{|l^T b|(1+\varepsilon)} \right\}, \quad \text{sign}(u_k^L) = -\text{sign} \left(\frac{l^T A x_k}{l^T b} \right).$$

В таблице 1 приведены числовые значения параметров системы.

Таблица 1. Числовые значения параметров.

N	φ	ε	δt	m_x	σ_x	σ_η
100	0,95	0,4	1	(0,1 0,01)	(0,1 -0,9)	(0,09 0,1)

На рисунках 1 и 2 приведены стохастические траектории координат вектора состояния при использовании указанных выше законов управления.

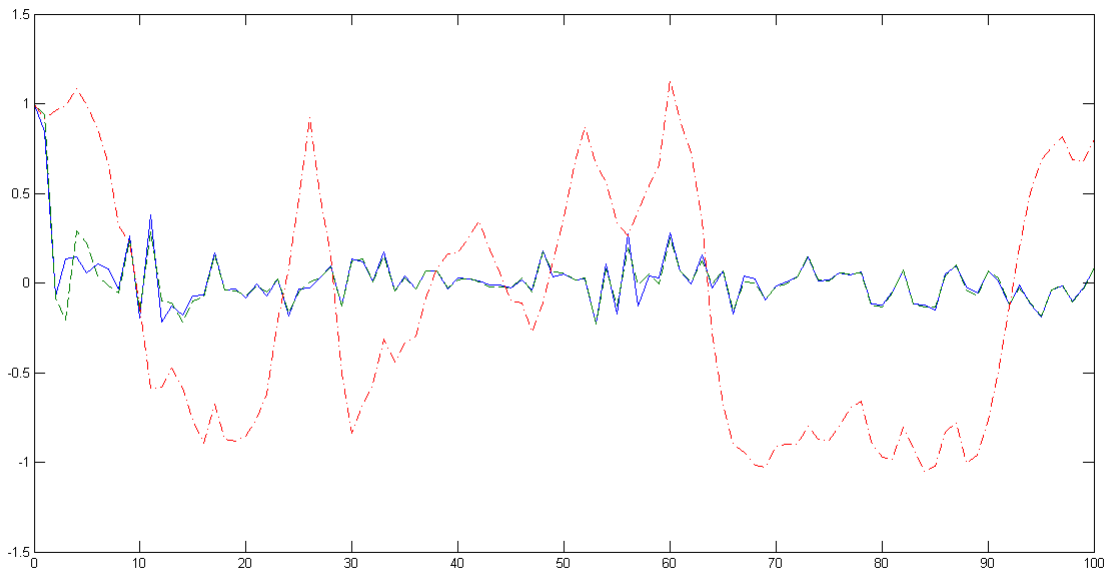


Рис. 1. – Траектория отклонения ГДВУ x_k^1
 (сплошная линия– при использовании управления u_k^L , штрих-пунктир - u_k^O ,
 пунктир - u_k^M).

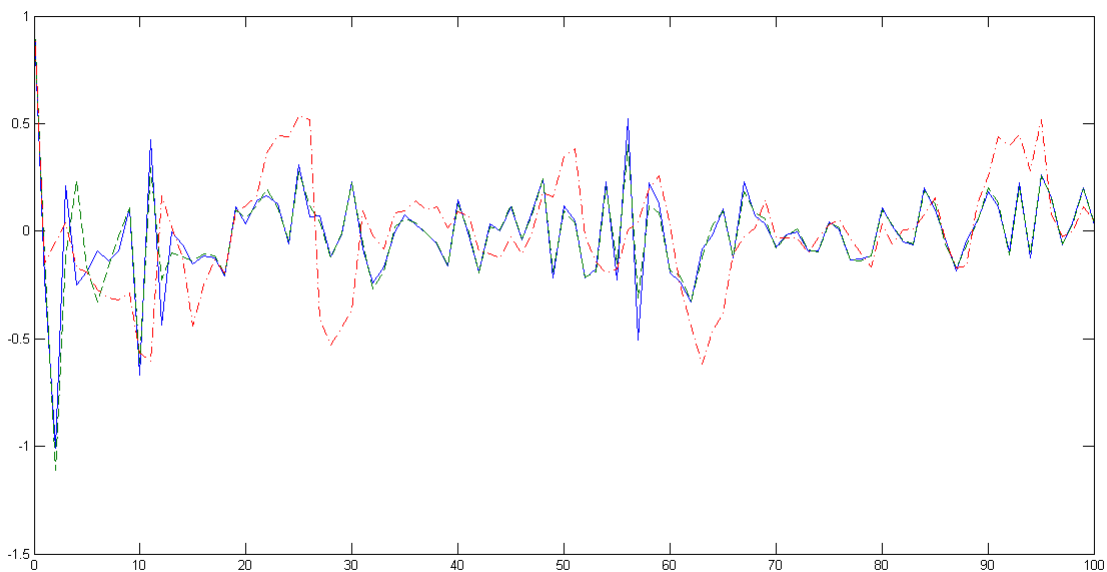


Рис. 2. – Траектория скорости дрейфа x_k^2
 (сплошная линия– при использовании управления u_k^L , штрих-пунктир - u_k^O ,
 пунктир - u_k^M).

Ниже приведены значения вероятностного критерия для указанных стратегий

$$P_\varphi(u^L(\cdot)) = P_\varphi(u^M(\cdot)) = 1, \quad P_\varphi(u^O(\cdot)) = 0.$$

Видно, что использовать управление с зоной нечувствительности при наличии аддитивных помех абсолютно неэффективно. Несмотря на это, управление с линейным участком, найденное без учета аддитивных возмущений, оказалось «робастным» к

последним, даже при «достаточно большом» уровне их дисперсий, что обосновывает его использование в задаче удержания ИСЗ в окрестности круговой орбиты.

6. Заключение

Рассмотрена задача оптимизации вероятности пребывания билинейной дискретной стохастической системой в окрестности нуля. На основе МДП и его модификации, разработанной в настоящей работе, для поверхности уровня 1 функции Беллмана, приведено аналитическое решение задачи для некоторого класса систем. На примере задачи удержания ИСЗ в окрестности круговой орбиты проведена проверка эффективности полученных результатов.

Литература

1. Малышев В.В., Кибзун А.И. Анализ и синтез высокоточного управления летательными аппаратами. М.: Машиностроение, 1987.
2. Малышев В.В. Задача об оптимальном дискретном управлении конечным состоянием линейной стохастической системы. // *АиТ*. 1967. №5. С 64-70.
3. Малышев В.В., Красильщиков М.Н., Бобронников В.Т., Нестеренко О.П., Федоров А.В. Спутниковые системы мониторинга. - М.: Изд-во МАИ, 2000.
4. Азанов В.М., Кан Ю.С. Однопараметрическая задача оптимальной коррекции траектории летательного аппарата по критерию вероятности // *Известия РАН. ТиСУ*. 2016. №2. С. 1-13.
5. Кан Ю.С. Об обосновании принципа равномерности в задаче оптимизации вероятностного показателя качества. // *АиТ*. 2000. №1. С 54-70.
6. Barmish B. R., Lagoa C. M. The uniform distribution: a rigorous justification for its use in robustness analysis // *Math. Control, Signals, Systems*. 1997. V. 10. P. 203–222.
7. Малышев В.В., Старков А.В., Федоров А.В. Синтез оптимального управления при решении задачи удержания космического аппарата в орбитальной группировке. // *Космонавтика и ракетостроение*. 2012. №4. С 150-158.
8. Малышев В.В., Старков А.В., Федоров А.В. Совмещение задач удержания и уклонения в окрестности опорной геостационарной орбиты. // *Вестник Московского городского педагогического университета. Серия: экономика*. 2013. №1. С 68-74.
9. Красильщиков М.Н., Малышев В.В., Федоров А.В. Автономная реализация динамических операций на геостационарной орбите. I. Формализация задачи управления // *Известия РАН. ТиСУ*. 2015. №6. С. 82 -98.

ОБ ОЦЕНКЕ ЭФФЕКТИВНОСТИ НЕЙРОСЕТЕВОГО НАСТРОЙЩИКА ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА ПРИ УПРАВЛЕНИИ НАГРЕВАТЕЛЬНЫМ ОБЪЕКТОМ В УСЛОВИЯХ ДЕЙСТВИЯ ВОЗМУЩЕНИЙ

Глуценко Антон Игоревич

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВО
Национальный исследовательский технологический университет "МИСиС"
309516, Белгородская обл., г. Старый Оскол, м-н Макаренко, д.42, strondutt@mail.ru

Фомин Андрей Вячеславович

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВО
Национальный исследовательский технологический университет "МИСиС"
309516, Белгородская обл., г. Старый Оскол, м-н Макаренко, д.42, verner444@yandex.ru

Еременко Юрий Иванович

Аннотация

В статье рассмотрена проблема оперативной подстройки параметров ПИ-регулятора как для компенсации нелинейности нагревательных печей, так и для обработки возмущающих воздействий, влияющих на них. Такую адаптацию предлагается производить с помощью нейросетевого настройщика параметров ПИ-регулятора, на настоящий момент включающего две нейронные сети, отвечающих за выбор значений параметров K_p и K_i для режимов нагрева и охлаждения соответственно. В рамках данного исследования в структуру настройщика добавлена еще одна нейронная сеть для подстройки параметров регулятора в условиях действия ступенчатых и импульсных возмущений, влияющих на выход объекта управления. Разработаны усовершенствованная структура настройщика и алгоритм его работы. Доработана база правил, определяющая моменты, когда необходимо вести обучение новой нейронной сети, а также скорости обучения отдельных нейронов. Результаты натурных экспериментов позволяют сделать вывод о сокращении времени на обработку возмущающих воздействий в среднем на 20% при использовании предлагаемого настройщика по сравнению с контуром управления с обычным ПИ-регулятором.

1. Введение

На сегодняшний день системы автоматического управления технологическими процессами достигли в своем развитии такого уровня, когда простая замена существующих контроллеров на более современные уже не приносит прироста показателей эффективности ведения контролируемого процесса. Это связано с тем, что алгоритм управления, заложенный в такие контроллеры, остается неизменным – это в большинстве случаев линейный ПИ- или ПИД-алгоритм [1]. В то же время, реальные объекты управления являются нелинейными. Это несоответствие приводит к тому, что параметры регулятора, подобранные для конкретного режима работы такого объекта, в других режимах не позволяют получать столь же высокого качества переходных процессов, что в конечном итоге выражается в повышенном расходе энергоносителя на ведение контролируемого процесса. Эта проблема особенно актуальна для несимметричных объектов [2, 3], таких как нагревательные печи, поскольку для них наличие даже небольшого перерегулирования по итогам переходного процесса потребует значительного времени на охлаждение [4].

Данная проблема может быть решена путем постоянной подстройки параметров применяемого в контуре управления линейного регулятора средствами адаптивной системы управления [1, 2, 5, 6]. Разработанный для этих целей настройщик ПИД-регуляторов будет иметь широкие перспективы внедрения, поскольку, в соответствии с [1, 5], более 95% контуров управления на реальных производствах основаны именно на ПИД-алгоритме.

Рассмотрим подробнее задачу разработки такого настройщика для управления нагревательными печами. Данный объект, как уже упоминалось выше, является несимметричным, т.е. требует эффективного управления двумя различными по скорости протекания процессами: нагреванием и охлаждением. В [2, 3] показано, что в таком случае для ПИД-регулятора должны быть подобраны два набора параметров. Кроме того, на объект действуют возмущения различной природы, которые должны

отрабатываться своим отдельным набором параметров регулятора. Таким образом, задача заключается в подборе и постоянной актуализации трех наборов параметров регулятора. В дальнейшем необходимо рассмотреть методы настройки ПИД-регулятора, обзор большинства из которых подробно приведен в [5, 7, 8]. В целом, среди них можно выделить группу классических методов, таких как:

1) метод Циглера–Никольса [9] (требует или наличие модели объекта управления (ОУ), или вывод системы управления на границу устойчивости), в большинстве случаев позволяющий получать коэффициенты, значения которых достаточно далеки от оптимальных;

2) методы, основанные на построении модели ОУ с помощью специального испытательного сигнала (ступенчатого [10, 11, 12] или полигармонического [13, 14]): метод частотной идентификации и настройки А.Г.Александрова [13], метод, заложенный в основу работы адаптивных ПИД-регуляторов фирмы Siemens [4] (на подобном же принципе основаны адаптивные регуляторы фирм Omron, Control Station и др.) и пр. Подобная идентификация ОУ даже из сходных начальных условий может давать существенно отличающиеся друг от друга результаты.

Кроме того, эти методы предполагают полный отказ от используемых в настоящее время в регуляторе коэффициентов и подбор новых. Моменты времени, когда необходимо использовать метод адаптации, определяются человеком, как и амплитуда испытательного сигнала. Таким образом, применение данных методов для автоматической оперативной подстройки параметров регуляторов является затруднительным.

В то же время, опытный инженер АСУ ТП способен подбирать параметры ПИД-регулятора, не имея модели и не применяя тестовые сигналы, а используя свои знания, опыт и возможность предвидеть дальнейшее поведение процесса. Указанные характеристики в определенной мере позволяют воспроизводить интеллектуальные методы, что делает перспективным их применение для рассматриваемой задачи.

Методы роя частиц [15] и генетических алгоритмов [16, 17] применяются для решения задач оптимизации, причем изначально случайным образом генерируется исходное множество решений. В случае с задачей настройки параметров регулятора, каждое решение – это набор значений параметров K_p , K_i , K_d . В качестве целевой функции выступает некоторая оценка качества переходного процесса, получаемого при конкретных значениях параметров регулятора. Методы являются итерационными, то есть в регулятор необходимо последовательно подставлять все имеющиеся в популяции наборы параметров и, для оценки их оптимальности, проводить опыты на реальном объекте. Или иметь точную модель ОУ, что, как уже упоминалось выше, является достаточно сложной задачей.

Настройка ПИД-регулятора с помощью нечеткой логики [18-21] и экспертных систем [22] может осуществляться без использования модели ОУ. В частности, в экспертных системах в базе правил указано на сколько процентов следует изменить амплитудное значение того или иного коэффициента регулятора в зависимости от текущей ситуации, описанной в условии конкретного правила. Проблемой подобных подходов является отсутствие механизма обучения, поскольку первоначально степень нелинейности ОУ достаточно тяжело оценить. В связи с чем, в базу правил (в частности, в их следствия) ЭС необходимо будет вносить изменения уже в процессе работы. В случае с использованием нечеткой логики для создания настройщика параметров регулятора его входные и выходные переменные определяют в диапазоне [-1;1] или [0;1]. Реальные же измеряемые на ОУ значения параметров затем приводятся в данные диапазоны с помощью коэффициентов нормировки. Также поступают с получаемыми с выходов настройщика значениями K_p , K_i , K_d (их переводят из

промежутка $[0;1]$ в диапазон, требуемый для данного ОУ). Данные коэффициенты нормировки также необходимо подстраивать в процессе работы [20].

Нейронные сети [23-29] (НС) могут быть обучены оперативно, что позволяет снизить требования к точности первоначальной оценки степени нелинейности объекта управления и поддерживать настройщик адекватным текущему состоянию контролируемого процесса. Однако НС не обладают экспертными знаниями, например, о том, что объект является несимметричным, или об ограничениях, накладываемых на сигнал управления и пр.

Нейросетевой настройщик параметров ПИ-регулятора, предложенный в [30, 31] сочетает в себе преимущества указанных выше подходов и позволяет избежать их недостатков. Он содержит в своем составе базу правил, основанную на опыте эксперта-настройщика регуляторов, определяющую когда необходимо обучать нейронную сеть и какую скорость следует использовать для отдельных ее нейронов. Кроме того, настройщик включает саму нейронную сеть (или несколько сетей), обучаемую оперативно и вычисляющую на своих выходах значения параметров регулятора K_p и K_i . Однако в указанных работах приведены принципы построения такого настройщика для эффективного отслеживания последовательности ступенчато сменяющих друг друга значений уставки. При этом нагревательная печь может менять свои состояния не только в результате смены задания или ее загрузки/разгрузки, но и ввиду действия импульсных и ступенчатых возмущений на ее выход. Целью данной работы является усовершенствование нейросетевого настройщика для придания ему возможности подбора параметров регулятора для отработки таких возмущений. В качестве объекта управления использована муфельная электронагревательная печь.

2. Постановка задачи

Предметом данного исследования является система управления, изображенная на рис.1. В качестве объекта управления выступает нагревательная печь, которая может быть описана как два апериодических звена первого порядка (модель самой печи и термопары) с запаздыванием (1).

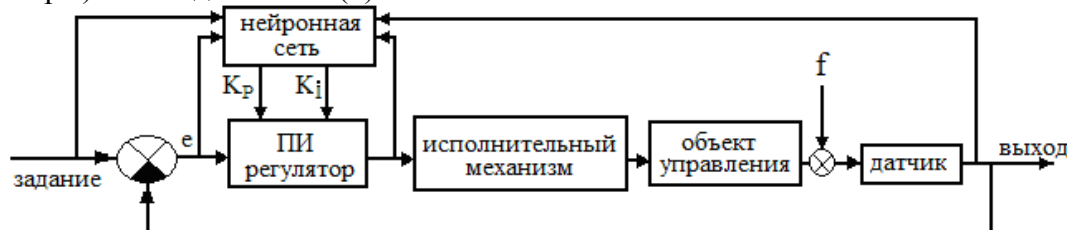


Рис. 1. – Система управления с нейросетевым настройщиком.

$$W(s) = (K / (T_1 s + 1)) \cdot (1 / (T_2 s + 1)) \cdot e^{-\tau \cdot s} \quad (1)$$

где K – коэффициент усиления, T_1 и T_2 – постоянные времени печи и термопары соответственно, τ – время запаздывания, f – ступенчатое или импульсное возмущающее воздействие (вызванное, например, открыванием печи или постоянной утечкой тепла, т.е. $f \leq 0$). Рассматриваемый объект является нелинейным, поскольку в действительности T_1 и K не являются константами, а меняют свои значения в соответствии с изменением режима работы объекта. Постоянная времени печи может изменяться в зависимости от количества материала, загруженного в эту печь. Коэффициент усиления уменьшается по мере увеличения температуры в печи, а кроме того, зависит от качества энергоносителя, подаваемого на печь.

Возмущение возникает в моменты, когда технологический процесс находится в установившемся режиме. Ступенчатое возмущение существует в течение нескольких смен уставки, а затем его амплитуда приравняется к нулю. Амплитуда импульсного возмущения приравняется к нулю в тот момент, когда переходный процесс, вызванный данным возмущением, достигает своего минимального значения. Все указанные выше причины и обуславливают необходимость подстройки параметров ПИ-регулятора.

Предполагается, что объект уже функционирует под управлением ПИ-регулятора, параметры которого не изменялись с течением времени. Они позволяют системе стабильно работать, но качество переходных процессов не является оптимальным. Задание реализовано в виде последовательности ступенчатых смен значений уставки. Система управления должна отслеживать такие изменения. Задача заключается в минимизации времени переходного процесса при сохранении требуемого его качества. Это позволит повысить энергетическую эффективность технологического процесса. Для решения данной задачи параметры K_p и K_i должны подстраиваться в соответствии с изменениями T_1 и K и при возникновении возмущений.

Решать поставленную задачу предлагается с помощью нейросетевого настройщика параметров ПИ-регулятора без построения модели объекта управления. Текущая ситуации на объекте управления анализируется с помощью базы правил, а затем принимается решение: следует ли в текущий момент обучать нейронную сеть и, если да, то с какой скоростью.

3. Описание нейросетевого настройщика

Предлагаемая функциональная схема нейросетевого настройщика приведена на рис.2. Одна из нейронных сетей используется для вычисления параметров регулятора при обработке возмущающих воздействий, две оставшиеся нейронные сети – для решения той же задачи для процессов нагрева и охлаждения печи соответственно. Какая нейронная сеть должна быть использована в текущий момент, определяется в соответствии со значениями задания и выхода объекта. Если задание было изменено и его предыдущее значение меньше текущего, то необходимо использовать нейронную сеть для процессов нагрева, иначе – для остывания. Если технологический процесс достиг установившегося режима, а затем контролируемая величина отклоняется от значения уставки больше допустимого, значит, возникло возмущающее воздействие – требуется использовать третью нейронную сеть.

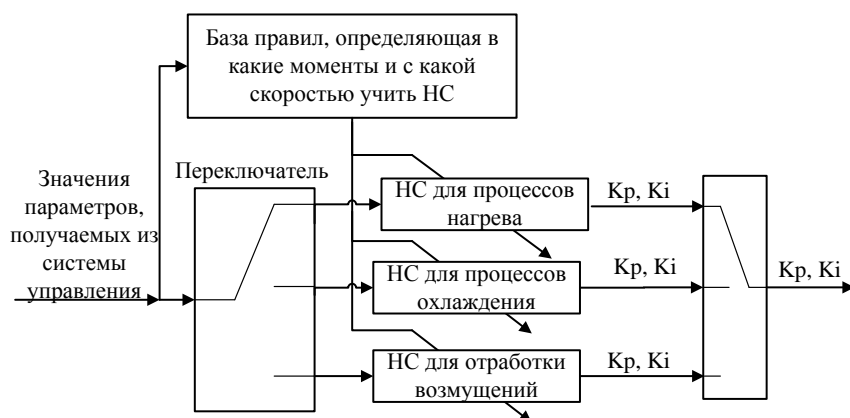


Рис. 2. – Функциональная схема нейросетевого настройщика.

Все нейронные сети имеют одинаковую трехслойную архитектуру, поскольку

любая непрерывная функция может быть аппроксимирована с помощью такой структуры [32]. Все параметры сети были определены экспериментальным путем в ходе исследований [30, 31, 33], результаты которых совпали с [23]. Во входном слое расположено 5 нейронов: текущее задание (x_1); задержанный на 1 секунду (x_2), на Δt секунд (x_3) и на $2 \cdot \Delta t$ секунд (x_4) сигнал по выходу ОУ; задержанный на 1 секунду сигнал управления, формируемый ПИ-регулятором (x_5). Входные сигналы нормируются в интервал $[0;1]$. Обращение к НС также происходит раз в Δt секунд. В скрытом слое расположено 15 нейронов с сигмоидальной функцией активации. Выходной слой состоит из двух нейронов с линейной функцией активации, отвечающих за вычисление параметров регулятора K_p и K_i .

Алгоритм работы настройщика приведен на рис. 3

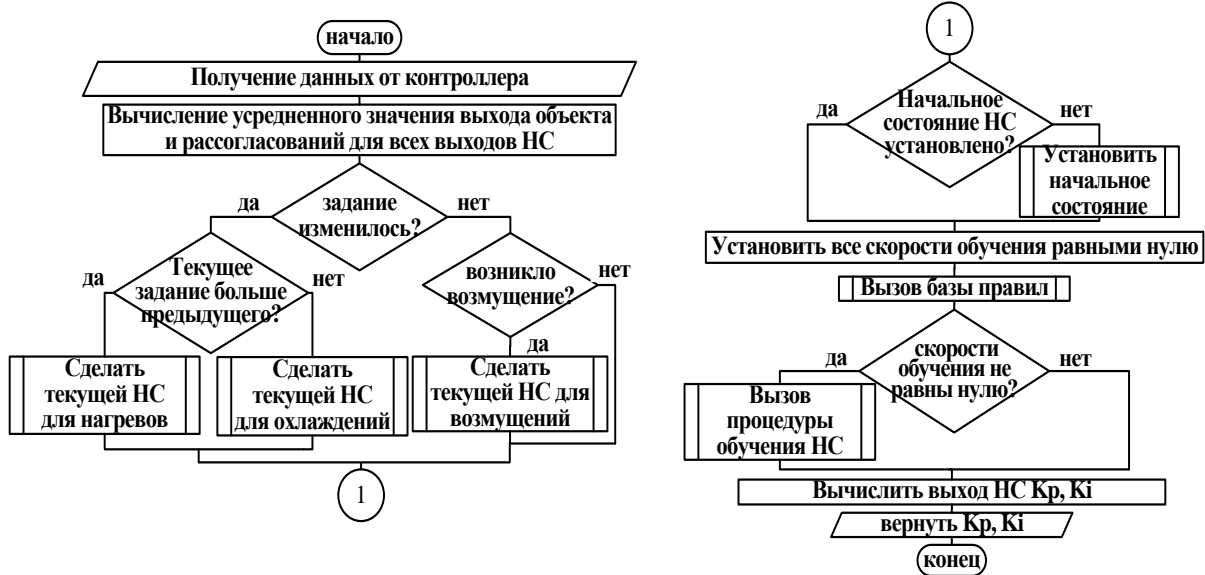


Рис. 3. – Алгоритм работы нейросетевого настройщика

В начальный момент времени выходной сигнал нейронной сети должен совпадать со значениями параметров регулятора, использовавшихся до интеграции настройщика в систему управления. Метод достижения данной цели описан в [31] и основан на результатах, приведенных в [34]. Он позволяет задавать «начальное состояние» для всех нейронных сетей, не требуя их предварительного автономного обучения классическими методами.

Для дальнейшего оперативного обучения нейронных сетей используется метод обратного распространения ошибки [23]. Формулы для коррекции весовых коэффициентов и смещений нейронов выходного слоя представлены в виде (2).

$$\begin{aligned} \Delta \omega_{kj}^{(2)}(t) &= \eta_k^{(2)} \delta_k^{(2)} O_j^{(2)} \\ \Delta b_k^{(2)}(t) &= \eta_k^{(2)} \delta_k^{(2)} \\ \delta_k^{(2)} &= e_k \frac{dO_k^{(2)}}{ds_k^{(2)}} = e_k \cdot 1 = e_k, k = \overline{1, 2}. \end{aligned} \quad (2)$$

$$e_2(t) = r(t) - y(t),$$

$$e_1(t) = e_2(t) - e_2(t-1).$$

$r(t)$ – это задание по температуре; $y(t)$ – выход ОУ; $\eta_k^{(2)}$ – скорость обучения для k -го нейрона выходного слоя; $\delta_k^{(2)}$ – совокупная ошибка k -го нейрона выходного слоя; e_k – ошибка k -го нейрона выходного слоя; $dO_j^{(2)}/ds_j^{(2)}$ – производная функции активации выходного слоя.

В данном случае предлагается использовать различные значения скоростей обучения для каждого из нейронов выходного слоя сети $\eta_1^{(2)}$ и $\eta_2^{(2)}$, а также менять значения этих скоростей в процессе функционирования. Это вызвано, во-первых, тем фактом, что порядок значений самих параметров K_p и K_i может отличаться (например, $K_p = 2.5$, а $K_i = 1.67 \cdot 10^{-2}$). Кроме того, в определенных случаях настройки требует только один параметр из двух, а иногда настройка не требуется совсем.

В начале каждой итерации работы нейросетевого настройщика (вне зависимости от того, какая нейронная сеть используется) все скорости обучения приравняются к нулю. Затем производится вызов базы правил. Условия правил описывают ситуации, когда нейронную сеть необходимо обучать. Следствия правил содержат значения скоростей обучения для нейронов скрытого слоя и каждого из нейронов выходного слоя. База правил для отработки переходных процессов приведена в [31]. Если же возникает возмущающее воздействие, то вызываются правила, описанные ниже.

Таким образом, модель объекта управления не строится и не используется.

4. База правил для отработки возмущающих воздействий

а. Идентификация возмущения

Если переходный процесс уже достиг установившегося режима, а затем модуль отклонения выхода объекта от уставки оказался выше допустимого, то возникло возмущающее воздействие. В этом случае используемые в настоящий момент параметры нейронной сети сохраняются. Производится отработка возмущения с использованием настоящей базы правил. А после смены задания сохраненные параметры сети вновь записываются в нейронную сеть, а текущие – отбрасываются.

б. Правила для определения скорости обучения нейрона выходного слоя, ответственного за вычисление K_p

ЕСЛИ пик переходного процесса, вызванного возмущением, еще не достигнут, и значение модуля скорости изменения выхода объекта управления выше, чем амплитуда шума, измеренная в течение $2 \cdot \Delta t$ секунд в установившемся режиме, И значение модуля рассогласования больше 5% модуля разницы между текущей уставкой и предыдущей, ТОГДА скорость обучения нейрона выходного слоя $\eta_1^{(2)}$, ответственного за вычисление K_p , должна быть на два порядка меньше самого значения K_p . Это приведет к увеличению K_p с целью минимизации разницы между уставкой и амплитудным значением пика данного переходного процесса.

ЕСЛИ пик переходного процесса, вызванного возмущением, уже был достигнут, И текущее значение температуры выше, чем измеренное Δt секунд назад, И значение модуля скорости нарастания температуры больше, чем отношение разницы уставки и пика переходного процесса к величине Δt , ТОГДА действующее возмущение является импульсным (поскольку скорость нарастания температуры выше допустимой) и значение K_p должно быть уменьшено. Скорость обучения нейрона выходного слоя $\eta_1^{(2)}$, ответственного за вычисление K_p , должна быть на два порядка меньше самого значения K_p . Если данное правило не сработало, то возникшее возмущение было ступенчатым, уменьшение K_p не требуется.

в. Правила для определения скорости обучения нейрона выходного слоя, ответственного за вычисление K_i

ЕСЛИ пик переходного процесса, вызванного возмущением, достигнут, И значение модуля скорости изменения выхода объекта управления ниже, чем амплитуда шума, измеренная в течение $2 \cdot \Delta t$ секунд в установившемся режиме, И значение модуля

рассогласования больше 5% модуля разницы между текущей уставкой и предыдущей, ТОГДА скорость обучения нейрона выходного слоя $\eta_2^{(2)}$, ответственного за вычисление K_i , должна быть на пять порядков меньше самого значения K_i . Это приведет к увеличению K_i с целью минимизации возникшей статической ошибки.

5. Реализация нейросетевого настройщика для лабораторной печи

В качестве объекта управления была выбрана муфельная электронагревательная печь SNOL 40/1200. Система управления данной печью состояла из трех уровней (рис.4).

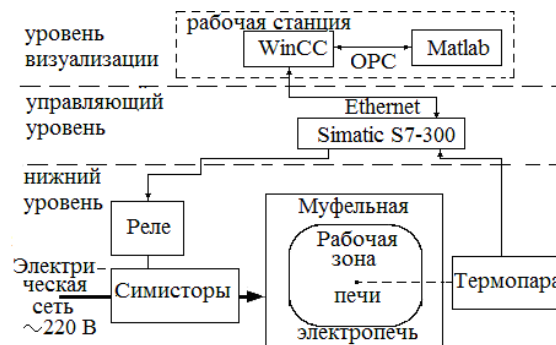


Рис.4. - Функциональная схема системы управления электрической печью

На нижнем уровне находились термопара, измеряющая температуру рабочей зоны печи, и реле, посредством симисторов включающее нагревательный элемент. Управляющий уровень (Simatic S7-300) осуществлял непосредственное управление муфельной электропечью. Контроллер получал данные от термопары через термопреобразователь, производил их обработку и отправлял на рабочую станцию, от которой он получал значение управляющего воздействия. В контроллере производилась широтно-импульсная модуляция управляющего воздействия и его выдача на исполнительный механизм включения нагревательного элемента.

На уровне визуализации находилась рабочая станция с системой визуализации WinCC и пакетом Matlab, в котором реализован нейросетевой настройщик параметров ПИ-регулятора. WinCC и Matlab взаимодействовали между собой через OPC-интерфейс посредством тегов. Схема управления, реализованная в пакете Matlab, приведена на рис.5. WinCC передавала данные о температуре (*temperature*) в Matlab. Значение уставки (*setpoint*) и выработанное регулятором управляющее воздействие (*control action*) передаются в WinCC, а оттуда – в контроллер. *Group1*, *group2* – это OPC блоки для соответственно записи и чтения значений в тегах WinCC.

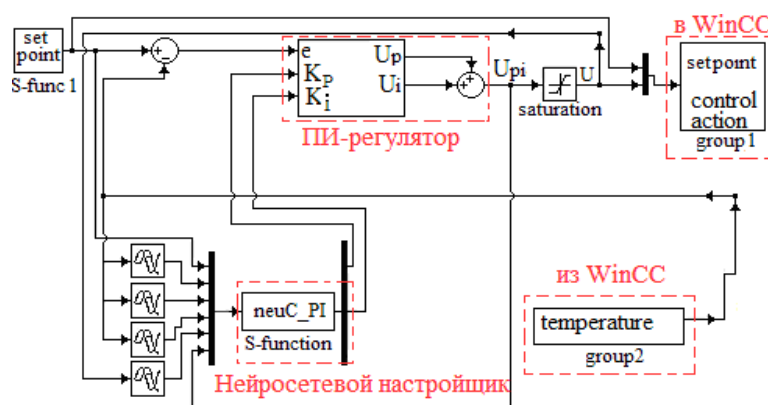


Рис.5. - Реализация системы управления с нейросетевым настройщиком в Matlab

В Matlab реализован ПИ-регулятор и настройщик (см. блок *S-function neuC_PI*, рис. 5). U_i , U_p , U_{pi} – сигналы управления на выходе i -, p -каналов и всего ПИ-регулятора соответственно. Блок *saturation* использован для ограничения сигнала управления с ПИ-регулятора до диапазона $[0;100]$ единиц. U – управляющее воздействие, подаваемое на ОУ.

Первые пять входов S-функции *neuC_PI* совпадают с описанными выше входами нейронных сетей. Шестой вход необходим для обеспечения работы базы правил, также реализованной внутри данной S-функции.

Изменение состояния печи производилось путем загрузки в нее литой заготовки, занимающей примерно 30% ее внутреннего пространства. В частности, эксперименты проводились следующим образом: пустая печь (12 смен задания) → загруженная печь (12 смен задания) → пустая печь (12 смен задания). График смены уставок представляет собой уменьшенный в два раза по амплитуде график заданий для печи нагрева литой заготовки перед прокатом, применяемый на металлургических предприятиях: $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$. Он реализован в виде S-функции *setpoint* (рис. 5). Для смены уставки температура должна удерживаться в окрестности задания, равной 5% модуля разницы между текущим заданием и предыдущим, в течение 30 минут.

Ступенчатые возмущающие воздействия были организованы в виде открывания дверцы печи в тот момент, когда переходный процесс после шестой, 18-й и 30-й смен задания достигал установившегося режима. Дверца закрывалась после того, как переходный процесс после девятой, 21-й и 33-й смен задания достигал установившегося режима.

Импульсные возмущающие воздействия были организованы в виде открывания дверцы печи в тот момент, когда переходный процесс после шестой, девятой, 18-й, 21-й, 30-й и 33-й смен задания достигал установившегося режима. Дверца закрывалась в тот момент, когда переходный процесс, вызванный поданным возмущением, достигал своего минимального значения по температуре.

Параметры ПИ-регулятора были подобраны для пустой печи для переходного процесса $505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$: $K_p = 2.5$, $K_i = 1.67 \cdot 10^{-2}$. В начале каждого эксперимента ПИ-регулятор с нейросетевым настройщиком использовал их в качестве стартовых. Система управления с обычным ПИ-регулятором, с которой производилось сравнение полученных результатов, всегда использовала только эти значения. В соответствии с [33], значение Δt было вычислено как 10 секунд.

6. Результаты натурных экспериментов

Экспериментальные результаты, полученные для систем управления с обычным ПИ-регулятором и нейросетевым настройщиком для импульсных возмущений, приведены на рис.6-7.

Для системы с нейросетевым настройщиком время эксперимента составило 38.9 часа, расход электроэнергии – 19.79 кВт·ч, время на отработку возмущений – 0.37 часа, среднее значение амплитуды максимального отклонения от задания при возмущении – 11.75°C . Для системы с обычным ПИ-регулятором время эксперимента составило 39.52 часа, расход электроэнергии – 20.39 кВт·ч, время на отработку возмущений – 0.5 часа, среднее значение амплитуды максимального отклонения от задания при возмущении – 16.09°C . Таким образом, нейросетевой настройщик позволил добиться 1.6% экономии по общему времени эксперимента, 2.9% экономии по расходу электроэнергии. Экономия времени на отработку возмущающих воздействий – 26%.

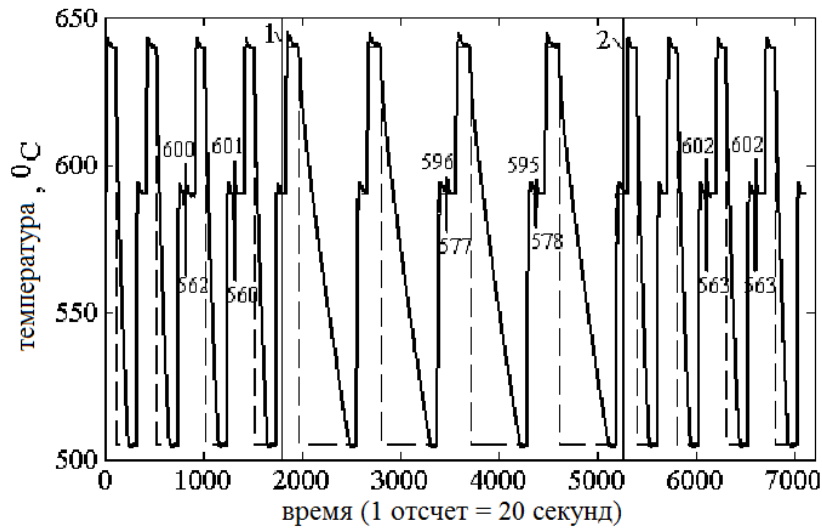


Рис. 6. – Результаты, полученные с помощью ПИ-регулятора для печи SNOL 40/1200 (1 – момент загрузки печи, 2 – момент разгрузки). Импульсные возмущения

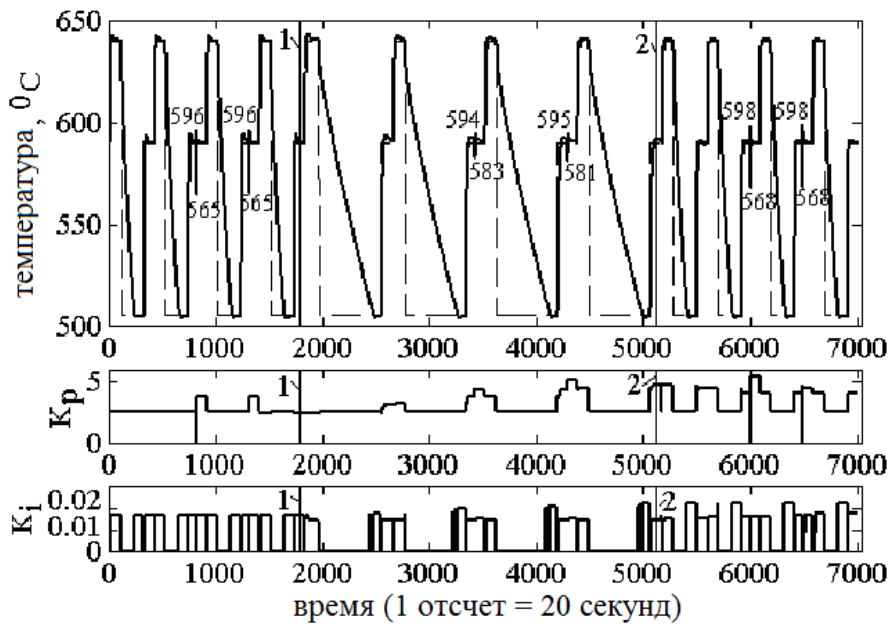


Рис. 7. – Результаты, полученные с помощью нейросетевого настройщика + ПИ-регулятор для печи SNOL 40/1200 (1 – момент загрузки печи, 2 – момент разгрузки). Импульсные возмущения

Экспериментальные результаты, полученные для систем управления с обычным ПИ-регулятором и нейросетевым настройщиком для ступенчатых возмущений, приведены на рис.8-9.

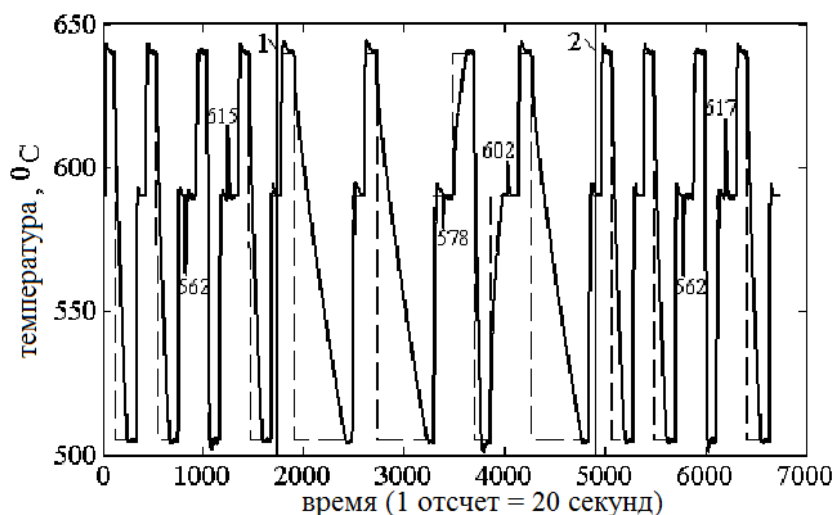


Рис. 8. – Результаты, полученные с помощью ПИ-регулятора для печи SNOL 40/1200 (1 – момент загрузки печи, 2 – момент разгрузки). Ступенчатые возмущения

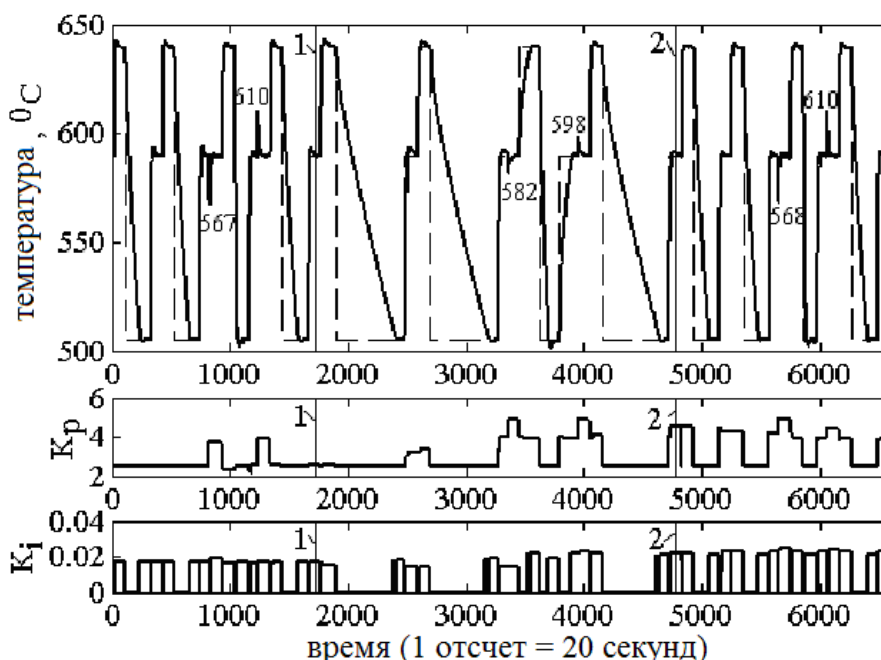


Рис. 9. – Результаты, полученные с помощью нейросетевого настройщика + ПИ-регулятор для печи SNOL 40/1200 (1 – момент загрузки печи, 2 – момент разгрузки). Ступенчатые возмущения

Для системы с нейросетевым настройщиком время эксперимента составило 36.53 часа, расход электроэнергии – 33.06 кВт·ч, время на отработку возмущений – 0.82 часа, среднее значение амплитуды максимального отклонения от задания при возмущении – 16.83°C . Для системы с обычным ПИ-регулятором время эксперимента составило 37.5 часа, расход электроэнергии – 35.5 кВт·ч, время на отработку возмущений – 1.03 часа, среднее значение амплитуды максимального отклонения от задания при возмущении – 22°C . Таким образом, нейросетевой настройщик позволил добиться 2.6% экономии по общему времени эксперимента, 6.9% экономии по расходу электроэнергии. Экономия времени на отработку возмущающих воздействий – 20%.

7. Заключение

Предлагаемые усовершенствования, внесенные в нейросетевой настройщик параметров ПИ-регулятора, позволили обеспечить стабильную работу НС и ее обучаемость в контуре управления в реальном масштабе времени при обработке возмущающих воздействий, имеющих ступенчатый и импульсный характер.

Дополнительная нейронная сеть и блок базы правил позволяют получать отдельный набор параметров регулятора для борьбы с возмущениями. Применение подобного настройщика для управления муфельной электронагревательной печью позволило сократить время на обработку возмущений минимум на 20%, а амплитуду отклонения от задания уменьшить минимум на 23.5%. Ввиду того, что на выполнение графика задания в обоих опытах система управления с нейросетевым настройщиком затратила меньше времени и электроэнергии, можно сделать вывод о том, что использование подобного оптимизатора позволило бы интенсифицировать производство на реальном объекте управления, что принесло бы экономический эффект. Дальнейшие исследования связаны с применением настройщика для обработки возмущений, подаваемых в канал управления.

Литература

1. Astrom K.J., Hagglund T. *Advanced PID Control*. – Research Triangle Park: The Instrumentation, Systems, and Automation Society, 2006. – 461 p.
2. Mimura K., Shiotsuki T. Experimental study of PID auto-tuning for unsymmetrical processes. *Proceedings of ICCAS-SICE 2009*. – IEEE, 2009. – P. 2967-2971.
3. Wei M., Luo X., Feng A. Predictive controller for nonlinear unsymmetrical systems based on control performance comparison // *CIESC Journal*. – 2012. – Vol. 63. – № 10. – P. 3183-3189.
4. Pfeiffer B.–M. Towards «plug and control»: self-tuning temperature controller for PLC // *International journal of Adaptive Control and Signal Processing*. – 2000. – № 14. – P. 519-532.
5. Astrom K.J., Wittenmark B. *Adaptive control*. Second Edition. – New York: Dover publications, 2008. – 590 p.
6. Тюкин И.Ю., Терехов В.А. Адаптация в нелинейных динамических системах. – М.: ЛКИ, 2008. – 384 с.
7. Li Y., Ang K., Chong C. Patents, software, and hardware for PID control – an overview and analysis of the current art // *IEEE Control Systems Magazine*. – 2006. – № 26(1). – P. 42–54.
8. Александров А.Г., Паленов М.В. Состояние и перспективы развития адаптивных ПИД регуляторов в технических системах // *Автоматика и телемеханика*. – 2014. – № 2. – С. 16-30.
9. Ziegler J., Nichols N. Optimum settings for automatic controllers // *Trans. ASME*. – 1942. – № 65. – P.759-768.
10. Шубладзе А.М., Кузнецов С.И. Автоматически настраивающиеся промышленные ПИ и ПИД регуляторы // *Автоматизация в промышленности*. – 2007. – № 2. – С. 15–17.
11. Kuhn U.A. A practice oriented setting rule for pid-controllers: The t-sum-rule // *Automatisierungstechnische Praxis*. – 1995. – Vol. 37. – P. 10-16.
12. Schaedel H.M. A new method of direct PID controller design based on the principle of cascaded damping ratios // *European Control Conference (ECC)*. – IEEE, 1997. – P. 1265–1271.

13. Alexandrov A.G., Palenov M.V. Self-tuning PID-I controller. Proceedings of the 18th IFAC World Congress. – IFAC, 2011. – P. 3635-3640.
14. Ротач В.Я. Кузицин В.Ф. Петров С.В. Настройка регуляторов по переходным характеристикам систем управления без их аппроксимации аналитическими выражениями // Автоматизация в промышленности. – 2009. – № 11. – С.9-12.
15. Allaoua B., Gasbaoui B., Mebarki B. Setting up PID DC motor speed control alteration parameters using particle swarm optimization strategy // Leonardo Electronic Journal of Practices and Technologies. – 2009. – Vol. 14. – P. 19–32.
16. Bindu R., Namboothiripad M.K. Tuning of PID controller for DC servo motor using genetic algorithm // International Journal of Emerging Technology and Advanced Engineering. – 2012. – Vol. 2. – № 3. – P. 310–314.
17. Unal M., Ak A., Topuz V., Erdal H. Optimization of PID Controllers Using Ant Colony and Genetic Algorithms. – London: Springer, 2013. – 85 p.
18. Кудинов Ю.И., Келина А.Ю. Упрощенный метод определения параметров нечетких ПИД регуляторов // Мехатроника, автоматизация, управление. – 2013. – № 1. – С. 12–22.
19. Erenoglu I., Eksin I., Yesil E., Guzelkaya M. An intelligent hybrid fuzzy PID controller. European Conference on Modelling and Simulation. – Bonn: European Council for Modelling and Simulation, 2006. – P. 62–67.
20. Guzelkaya M., Eksin I., Yesil E. Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer // Engineering Applications of Artificial Intelligence. – 2003. – Vol. 16(3). – P. 227–236.
21. Ahmed H., Rajoriya A. Performance Assessment of Tuning Methods for PID Controller Parameter used for Position Control of DC Motor // International Journal of u- and e-Service, Science and Technology. – 2014. – Vol. 7. – № 5. – P. 139-150.
22. Anderson K.L., Blankenship G.I., Lebow L.G. A rule-based adaptive PID controller. Proc. 27th IEEE Conf. Decision. Control. – IEEE, 1988. – P. 564-569.
23. Omatu S., Khalid M., Yusof R. Neuro-Control and its Applications. – London: Springer, 1995. – 255 p.
24. Chen J., Huang T. Applying neural networks to on-line updated PID controllers for nonlinear process control // Journal of Process Control. – 2004. – № 14. – P. 211-230.
25. Fang M., Zhuo Y., Lee Z. The application of the self-tuning neural network PID controller on the ship roll reduction in random waves // Ocean Engineering. – 2010. – № 37. – P. 529-538.
26. Reyes J., Astorga C., Adam M., Guerrero G. Bounded neuro-control position regulation for a geared DC motor // Engineering Applications of Artificial Intelligence. – 2010. – № 23. – P. 1398-1407.
27. Ferreira P.M., Faria E.A., Ruano A.E. Neural network models in greenhouse air temperature prediction // Neurocomputing. – 2002. – Vol. 43(1). – P. 51-75.
28. Васильев В.И., Идрисов И.И. Алгоритмы проектирования и анализа устойчивости интеллектуальной системы управления // Вестник Уфимского государственного авиационного технического университета. – 2008. – Т. 11. – № 1. – С. 34–42.
29. Макаров И.М., Лохин В.М. Интеллектуальные системы автоматического управления. – М.: ФИЗМАТЛИТ, 2001. — 576 с.
30. Eremenko Y.I., Poleshchenko D.A., Glushchenko A.I. On Applying Neural Tuner to PI-Controller Parameters Calculation for Heating Furnaces Control. Proc. IEEE 2015 International Siberian Conference on Control and Communications (SIBCON). – IEEE, 2015.
31. Еременко Ю.И., Полещенко Д.А., Глущенко А.И. О применении нейросетевого оптимизатора параметров ПИД-регулятора для управления нагревательными печами в

- различных режимах работы // Управление большими системами. – 2015. – Вып. 56. – С. 143-175.
32. Hornik K., Stinchcombe M., White H. Multilayer feed-forward networks are universal approximators // Neural networks. – 1989. – Vol. 2. – № 3. – P. 359-366.
33. Еременко Ю.И., Полещенко Д.А., Глущенко А.И. О разработке метода определения значений задержек входных сигналов нейронной сети при реализации нейросетевого оптимизатора параметров ПИ-регулятора // Известия ЮФУ. Технические науки. – 2015. – № 10. – С. 157-168.
34. Huang G.B., Wang D.H., Lan Y. Extreme learning machines: a survey // International Journal of Machine Learning Cybernetics. – 2011. – № 2. – P. 107-122.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант 15-07-0609215) и фонда Президента РФ (грант №14.У30.15.4865-МК).

РОБАСТНОСТЬ СЛАБО НЕЛИНЕЙНОЙ ДИСКРЕТНОЙ СИСТЕМЫ ПО ОТНОШЕНИЮ К ПАРАМЕТРИЧЕСКИМ ВОЗМУЩЕНИЯМ

Даник Юлия Эдуардовна

Институт системного анализа Федерального исследовательского центра «Информатика и управление» Российской академии наук
Москва, yuliadanik@gmail.com

Аннотация

Данная работа посвящена оценке робастности к параметрическим неопределенностям замкнутой дискретной квазилинейной системы вдоль нелинейного стабилизирующего «срединного» регулятора, построенного на основе дискретного уравнения Риккати с зависящими от состояния коэффициентами D-SDRE (Discrete-Time State Dependent Riccati Equation). Приводится утверждение о робастности и результаты численных экспериментов на примере модели перевернутого маятника, демонстрирующих асимптотическую устойчивость замкнутой системы при заданном изменении параметров неопределенности и выполнении условий теоремы.

1. Введение

В настоящей работе исследуется вопрос робастности нелинейной замкнутой квазилинейной дискретной системы вдоль нелинейного регулятора по отношению к одному классу параметрических неопределенностей. Здесь под робастностью подразумевается способность системы сохранять свойства устойчивости при наличии неопределенности в параметрах системы. В литературе есть значительное число работ, посвященных оценке свойств робастности линейных стационарных непрерывных и дискретных замкнутых систем, основанных на представлении области неопределенности для матрицы замкнутой системы в виде выпуклых многогранных матричных множеств с последующим анализом свойств, входящих в них матричных элементов [3]-[10]. В этих работах предлагаются различные наборы матричных неравенств, позволяющих установить асимптотическую устойчивость для всей совокупности получаемых замкнутых систем, с указанием соответствующей функции Ляпунова.

Основная идея этой работы состоит в применении методов оценки робастности для стационарных систем к квазилинейным дискретным системам. Это представляется

возможным в виду того, что присутствие только слабой нелинейности в системе позволяет делать предположения о робастности всей системы только на основе ее линейной части с наложением ряда условий и ограничений. Анализ системы ведется вдоль так называемого «срединного» регулятора, рассчитанного при средних значениях неопределенности, который будет описан ниже.

Исследование выполнено за счет гранта Российского научного фонда (проект №14-11-00692).

2. Описание модели

Будем рассматривать дискретную управляемую систему на полуоси, коэффициенты которой являются слабонелинейными по состоянию с параметром μ

$$\begin{aligned} x(t+1) &= A(x(t), \mu)x(t) + B(x(t), \mu)u(t) = \\ &= (A_0 + \mu A_1(x(t)))x(t) + (B_0 + \mu B_1(x(t)))u(t) = g(x, \mu) \\ x(0) &= x_0, \quad x(t) \in X \in R^n, \quad u \in R^r, \quad t = 0, 1, 2, \dots, \quad 0 < \mu \leq \mu_0, \end{aligned} \quad (1)$$

где μ_0 – некоторое заданное число, A_0, B_0 – заданные постоянные матрицы, $A_0, A_1(x) \in R^{n \times n}$, $B_0, B_1(x) \in R^{n \times r}$, X – некоторое ограниченное множество пространства состояний.

В качестве нелинейного управления $u(x, \mu)$, обеспечивающего асимптотическую устойчивость замкнутой системы (1) с функционалом качества

$$I(u) = \frac{1}{2} \sum_{t=0}^{\infty} (x^T(Q_0 + \mu Q_1(x))x + u^T R_0 u) \rightarrow \min \quad (2)$$

где Q_0, R_0 – постоянные матрицы, $Q_0 \geq 0, R_0 > 0$, и $Q_1(x) \geq 0$, возьмем следующий регулятор (см.[1])

$$u(x, \mu) = K(x, \mu)x = (K_0 + \mu K_1(x))x, \quad (3)$$

где $K_0, K_1(x)$ – некоторые постоянная и переменная матрицы соответственно.

Введем условие

I. Все коэффициенты матриц $A_1(x), B_1(x), Q_1(x)$ являются непрерывными ограниченными функциями в области X , $g(0, \mu) \equiv 0$ ($g(x, \mu) = A_{cl}(x(t), \mu)x(t)$) и параметр μ принадлежит ограниченному интервалу $(0, \mu_0]$.

Используемый алгоритм построения нелинейных стабилизирующих регуляторов вида (3) основан на формальном решении дискретного матричного алгебраического уравнения Риккати, но с коэффициентами, зависящими от состояния

$$\begin{aligned} A^T(x, \mu)P(x, \mu)A(x, \mu) - P(x, \mu) - \\ - A^T(x, \mu)P(x, \mu)B(x, \mu)(R_0 + B^T(x, \mu)P(x, \mu)B(x, \mu))^{-1}B^T(x, \mu)P(x, \mu)A(x, \mu) + Q(x, \mu) = 0, \end{aligned} \quad (4)$$

где $P(x, \mu) = P_0 + \mu P_1(x)$.

Последнее уравнение связано с условиями оптимальности в стационарной линейно-квадратичной задаче оптимального управления, но здесь уравнение используется как эвристическая конструкция. Аналогично, используется выражение для оптимального регулятора, связанного с решением уравнения Риккати

$$u(x, \mu) = - \left[R_0 + B^T(x, \mu)P(x, \mu)B(x, \mu) \right]^{-1} B^T(x, \mu)P(x, \mu)A(x, \mu)x \quad (5)$$

Для решения уравнения (4) используется разложение по параметру μ . Формально предполагая параметр малым, для обеспечения сходимости матричного ряда,

соответствующего обратной матрице $[R_0 + (B_0 + \mu B_1(x))^T (P_0 + \mu P_1(x))(B_0 + \mu B_1(x))]^{-1}$, приравниваем в (4) члены при одинаковых степенях μ и получаем систему матричных уравнений для определения членов $P_0, P_1(x)$ представления (5). В итоге уравнение для нахождения P_0 превращается в дискретное алгебраическое уравнение Риккати (6) относительно P_0 с матрицами A_0, B_0, R_0 , а $P_1(x)$ находится как положительно определенное решение дискретного уравнения Ляпунова (7)

$$A_0^T P_0 A_0 - P_0 - A_0^T P_0 B_0 (R_0 + B_0^T P_0 B_0)^{-1} B_0^T P_0 A_0 + Q_0 = 0 \quad (6)$$

$$A_{cl,0}^T P_1(x) A_{cl,0} - P_1(x) = -C(x), \quad (7)$$

где $\tilde{R}_0 = R_0 + B_0^T P_0 B_0$, $A_{cl,0} = A_0 - B_0 \tilde{R}_0^{-1} B_0^T P_0 A_0$ и

$$C(x) = A_0^T P_0 A_1(x) + A_1^T(x) P_0 A_0 - A_0^T P_0 B_1(x) \tilde{R}_0^{-1} B_0^T P_0 A_0 - A_0^T P_0 B_0 \tilde{R}_0^{-1} B_1^T(x) P_0 A_0 - \\ - A_1^T(x) P_0 B_0 \tilde{R}_0^{-1} B_0^T P_0 A_0 - A_0^T P_0 B_0 \tilde{R}_0^{-1} B_0^T P_0 A_1(x) + A_0^T P_0 B_0 \tilde{R}_0^{-1} B_0^T P_0 B_1(x) \tilde{R}_0^{-1} B_0^T P_0 A_0 + \\ + A_0^T P_0 B_0 \tilde{R}_0^{-1} B_1^T(x) P_0 B_0 \tilde{R}_0^{-1} B_0^T P_0 A_0 + Q_1(x).$$

Аналогично, подставляя $P_0, P_1(x)$ в (5) и формально предполагая μ малым можно разложить в ряд выражение для $u(x, \mu)$ и тогда, взяв для краткости только члены при нулевой и первой степенях μ получим

$$u(x, \mu) = u_0(x) + \mu \cdot u_1(x) = K_0 x + \mu K_1(x) x, \quad (8)$$

где

$$K_0 = -\tilde{R}_0^{-1} B_0^T P_0 A_0, \quad (9)$$

$$K_1(x) = \tilde{R}_0^{-1} \{ [B_0^T P_1(x) B_0 - B_0^T P_0 B_1(x) - B_1^T(x) P_0 B_0] \tilde{R}_0^{-1} B_0^T P_0 A_0 - \\ - [B_0^T P_0 A_1(x) + B_1^T(x) P_0 A_0 + B_0^T P_1(x) A_0] \} \quad (10)$$

Если учесть члены разложения $u(x, \mu)$ при высших степенях μ , очевидно, можно получить более точные результаты для регулятора. Утверждение об устойчивости системы, управляемой построенным нелинейным регулятором, по Ляпунову с функцией $V(x) = x^T P(x, \mu) x = x^T (P_0 + \mu P_1(x(t))) x$ описано в [1], а результаты численных экспериментов в [2].

В отличие от регулятора в стационарной линейно-квадратичной задаче, регулятор (8) является только субоптимальным по критерию (2), при этом матрицы критерия подбираются таким образом, чтобы было разрешимо уравнение (4) относительно $P_0, P_1(x)$.

Замкнутая система (1) вдоль управления (8) имеет вид

$$x(t+1) = (A_0 + \mu A_1(x(t)))x(t) + (B_0 + \mu B_1(x(t)))(K_0 + \mu K_1(x(t)))x(t) = \\ = [(A_0 + \mu A_1(x(t))) + (B_0 + \mu B_1(x(t)))(K_0 + \mu K_1(x(t)))]x(t) = A_{cl}(x(t), \mu)x(t), \quad x(0) = x^0 \quad (11)$$

3. Исследование свойств робастности замкнутой системы по отношению к параметрическим возмущениям

Теперь перейдем к оценке робастности замкнутой слабо нелинейной дискретной системы (11) относительно параметрических возмущений вдоль построенного нелинейного регулятора (8). Т.е. будем исследовать свойство грубости замкнутой системы, возможность сохранения устойчивости при наличии неопределенности в матрицах исходной системы.

В качестве параметров неопределенности будем рассматривать интервальные возмущения в коэффициентах замкнутой системы при $\mu = 0$, т.е. неопределенности в постоянных матрицах системы, не зависящих от состояния (A_0, B_0) . Параметры неопределенности $\theta = (\theta_1, \dots, \theta_p) \in R^p$ фиксированы и принимают значения из заданных ограниченных замкнутых интервалов $\theta \in \Omega = \{\theta : \underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, i = 1, \dots, p\}$. Пусть вектор с координатами $\theta_i^{cp} = \frac{\underline{\theta}_i + \bar{\theta}_i}{2}, i = 1 \dots p$ есть вектор средних значений параметров неопределенности.

Далее предполагаем, что матрицы $K_0, K_1(x)$ регулятора (8) рассчитываются при фиксированных значениях параметров неопределенности, здесь будут использованы средние значения. Назовем соответствующий регулятор «срединным», и будем использовать его при всех реализациях неопределенности θ в матрицах системы.

Срединный нелинейный регулятор имеет вид

$$u(x) = K_0(\theta^{cp})x(t) + \mu K_1(x, \theta^{cp})x, \quad (12)$$

где коэффициенты усиления $K_0, K_1(x)$ известны и рассчитываются по формулам (9) (10) при средних значениях параметров неопределенности θ^{cp} ,

$$\begin{aligned} K_0(\theta^{cp}) &= -(R_0 + B_0^T(\theta^{cp})P_0(\theta^{cp})B_0(\theta^{cp}))^{-1}B_0^T(\theta^{cp})P_0(\theta^{cp})A_0(\theta^{cp}), \\ K_1(x, \theta^{cp}) &= \tilde{R}_0^{-1}(\theta^{cp})\{[B_0^T(\theta^{cp})P_1(x, \theta^{cp})B_0(\theta^{cp}) - B_0^T(\theta^{cp})P_0(\theta^{cp})B_1(x) - \\ &- B_1^T(x)P_0(\theta^{cp})B_0(\theta^{cp})]\tilde{R}_0^{-1}(\theta^{cp})B_0^T(\theta^{cp})P_0(\theta^{cp})A_0(\theta^{cp}) - \\ &- [B_0^T(\theta^{cp})P_0(\theta^{cp})A_1(x) + B_1^T(x)P_0(\theta^{cp})A_0(\theta^{cp}) + B_0^T(\theta^{cp})P_1(x, \theta^{cp})A_0(\theta^{cp})]\} \end{aligned}$$

Будем называть регулятор, получаемый из (12) при $\mu = 0$

$$u(x) = K_0(\theta^{cp})x, \quad (13)$$

срединным линейным регулятором.

Введем условия

II. Тройка матриц $(A_0(\theta^{cp}), B_0(\theta^{cp}), Q_0^{\frac{1}{2}})$ является стабилизируемой и детектируемой.

III. Существует такая матрица $Q_1(x) > 0$, что $C(x, \theta^{cp})$ из (7), рассчитанная с матрицами $A_0(\theta^{cp}), B_0(\theta^{cp})$, является положительно определенной матрицей для любого $x \in X$.

Основная идея доказательства робастности заключается в том, что слабая нелинейность исходной системы позволяет изучать робастность замкнутой системы на основе свойств робастности соответствующей стационарной дискретной линейной системы при $\mu = 0$, а именно $x(t+1) = A_{cl,0}x(t)$, где $A_{cl,0}$ - стационарная матрица. В случае, если такая система робастна, то при условии малости μ сохраняется робастность и исходной системы. При этом предполагается, что параметры неопределенности таковы, что $A_{cl,0}$ может быть представлена в форме $A_{cl,0} = A_{cl,0}(\theta) = \tilde{A}_0 + \tilde{A}_1\theta_1 + \tilde{A}_2\theta_2 + \dots + \tilde{A}_p\theta_p$, где $\tilde{A}_k \in R^{n \times n}$, $k = 0, 1, \dots, p$ - постоянные матрицы. Такая зависимость $A_{cl,0}$ от компонентов вектора параметрических возмущений позволяет разделить параметры неопределенности в коэффициентах и учитывать интервалы их изменения. Выполнение этого предположения позволяет построить многогранную матричную область неопределенности и анализировать робастность замкнутой стационарной системы с применением линейных матричных

неравенств, решение которых легко найти с использованием математических пакетов, например, Matlab. Для случая матриц замкнутой системы, зависящих от состояния, как (1), аналогичных результатов по робастности в литературе в настоящий момент нет, ввиду трудности учета нелинейного характера изменения матриц системы.

Матрицу замкнутой системы (11) можно представить в следующем виде, разделив линейную и нелинейные части

$$\begin{aligned} x(t+1) &= [A_0 + \mu A_1(x(t)) + B_0 K_0 + \mu B_0 K_1(x) + \mu B_1(x) K_0 + \mu^2 B_1(x) K_1(x)] x(t) = \\ &= [A_0 + B_0 K_0 + \mu A_1(x(t)) + \mu B_0 K_1(x) + \mu B_1(x) K_0 + \mu^2 B_1(x) K_1(x)] x(t) = \\ &= A_{cl,0} x(t) + \mu [A_1(x(t)) + B_0 K_1(x) + B_1(x) K_0 + \mu B_1(x) K_1(x)] x(t), \end{aligned}$$

где $A_{cl,0} = A_0 + B_0 K_0$ - постоянная матрица, а остальные слагаемые входят со степенями параметра μ .

Сначала исследуем робастность относительно параметрических неопределенностей (входящих линейно) в матрицах A_0, B_0 замкнутой линейной невозмущенной (при $\mu = 0$) дискретной системы вдоль линейного срединного регулятора (13), т.е.

$$x(t+1) = A_{cl,0}(\theta) x(t) = (A_0(\theta) + B_0(\theta) K_0(\theta^{cp})) x(t), \quad (14)$$

Если система (14) будет робастна, то и $x(t+1) = A_{cl}(x, \mu, \theta) x(t)$ вдоль срединного нелинейного регулятора также окажется робастной при наложении ряда условий. Матрица нелинейной замкнутой системы с параметрами неопределенности вдоль срединного нелинейного регулятора имеет вид

$$A_{cl}(x, \mu, \theta) = A_{cl,0}(\theta) + \mu W(x, \mu, \theta), \quad (15)$$

где $A_{cl,0}(\theta) = A_0(\theta) + B_0(\theta) K_0(\theta^{cp})$,

$$W(x, \mu, \theta) = A_1(x(t)) + B_0(\theta) K_1(x(t), \theta^{cp}) + B_1(x(t)) K_0(\theta^{cp}) + \mu B_1(x(t)) K_1(x(t), \theta^{cp}).$$

Робастность, соответственно, будет зависеть от параметра μ и свойств матрицы $W(x, \mu, \theta)$.

Согласно [3] матрица замкнутой системы $A_{cl,0}(\theta)$, содержащая описанную выше неопределенность может быть представлена в виде многогранника Z , определенного конечным числом «крайних» матриц (вершин) A_1, \dots, A_N , $N = 2^p$

$$Z = \left\{ A_{cl,0}(\alpha) : A_{cl,0}(\alpha) = \sum_{i=1}^N \alpha_i A_i, \alpha \in \Delta_N \right\}, \quad \Delta_N = \left\{ \alpha \in R^N : \sum_{i=1}^N \alpha_i = 1 : \alpha_i \geq 0 \right\}, \quad (16)$$

и являющегося выпуклой оболочкой этих матричных элементов. В данном случае многогранник матриц, содержащий в себе все реализации матрицы $A_{cl,0}(\theta)$ при заданном диапазоне изменения параметров, порождается выпуклой оболочкой $N = 2^p$ вершин A_i , которые соответствуют различным комбинациям крайних значений (минимальных $\underline{\theta}_j$ или максимальных $\bar{\theta}_j$) координат вектора параметров неопределенности $\theta = \{\theta_i \in [\underline{\theta}_i, \bar{\theta}_i], i = 1, \dots, p$, подставленным в матрицу замкнутой системы $A_{cl,0}(\theta)$ из (14). То есть, можно говорить, что матрица замкнутой системы $A_{cl,0}(\theta)$ в (14) вдоль линейного срединного регулятора (13) принадлежит выпуклой ограниченной области Z (16) и при любой реализации неопределенности в матрице $A_{cl,0}(\theta)$, найдется вектор $\alpha = [\alpha_1, \dots, \alpha_N]^T$, которому соответствует матрица из множества Z , совпадающая с $A_{cl,0}(\theta)$.

Теперь, при заданном множестве Z необходимо установить, определяет ли каждый его матричный элемент асимптотически устойчивую дискретную систему

$$x(t+1) = A_{cl,0}(\alpha)x(t), \quad (17)$$

т.е. находятся ли собственные числа соответствующих матриц внутри единичного круга.

Для этого введем условие

IV. Найдутся положительно определенные матрицы $P_i, i=1, \dots, N$, такие что выполняется следующая система неравенств [5]

$$A_i^T P_i A_i - P_i < -E, \quad i=1, \dots, N, \\ A_i^T P_i A_j + A_j^T P_i A_i + A_i^T P_j A_i - 2P_i - P_j < \frac{1}{(N-1)^2} E, \quad i=1, \dots, N, \quad j \neq i, \quad j=1, \dots, N, \quad (18)$$

$$A_j^T P_i A_k + A_k^T P_i A_j + A_i^T P_j A_k + A_k^T P_j A_i + A_j^T P_k A_i - 2(P_i + P_j + P_k) < \frac{6}{(N-1)^2} E, \\ i=1, \dots, N-2, \quad j=i+1, \dots, N, \quad k=j+1, \dots, N$$

Отметим, что согласно [5] при выполнении условия IV для любой $A_{cl,0}(\alpha) \in Z$, положительно определенная квадратичная форма

$$\tilde{V}(x) = x^T P(\alpha)x \quad (19)$$

является положительно определенной функцией Ляпунова, зависящей от параметра,

такой что $A_{cl,0}(\alpha)^T P(\alpha) A_{cl,0}(\alpha) - P(\alpha) < 0$, где $P(\alpha) = \sum_{i=1}^N \alpha_i P_i$, $\sum_{i=1}^N \alpha_i = 1$, $\alpha_i \geq 0$.

Возможные модификации систем неравенств, для нахождения P_i и их сравнения приведены в [10].

Перейдем к определению достаточных условий, при которых имеет место робастная устойчивость нелинейной замкнутой системы с матрицей (15) по отношению к параметрическим возмущениям в матрице $A_{cl,0}$ (т.е. в матрицах A_0, B_0). Асимптотическая устойчивость сохраняется при переходе от линейной системы (с матрицей $A_{cl,0}(\theta)$, описываемой многогранником матриц $A_{cl,0}(\alpha)$) к нелинейной системе с матрицей (15) с той же функцией Ляпунова (19) для некоторого достаточно малого μ_0 . Установлена

Теорема. При выполнении условий I, II, III, IV существует достаточно малое $\mu_0 > 0$, что для любого $0 < \mu \leq \mu_0$ вдоль срединного нелинейного регулятора (12) замкнутая система с матрицей (15) робастна по отношению к параметрическим неопределенностям $\theta \in \Omega$ в матрицах A_0, B_0 .

Отметим, что, так как $A_1(x), B_1(x)$ входят в замкнутую систему с малым параметром, теорему можно обобщить на случай ограниченных параметрических неопределенностей, входящих в матрицы $A_1(x), B_1(x)$. Утверждение теоремы справедливо для достаточно малых $\mu > 0$, а по отношению к произвольным $\mu > 0$ носит эвристический характер.

Рассмотрим выполнение приведенного утверждения на примере задачи стабилизации перевернутого маятника [2].

$$x(t+1) = \begin{bmatrix} 1 & T_s \\ \frac{T_s g \sin(x_1)}{Lx_1} & 1 - \frac{T_s \gamma}{ML} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad (20)$$

где $T_s = 0.05$ – шаг дискретизации, который здесь используется как естественный малый параметр μ , $M = 0.1$, $L = 0.1$, $g = 9.8$, $\gamma = 0.05$.

Согласно приведенному выше алгоритму, преобразуем модель к виду (1). Пусть матрицы A_0, A_1, B_0, B_1 определяются как $A_0 = \begin{bmatrix} 1 & T_s \\ \frac{T_s g}{L} & 1 - \frac{T_s \gamma}{ML} \end{bmatrix}$, $A_1(x) = \begin{bmatrix} 0 & 0 \\ \frac{g \sin(x_1)}{Lx_1} - \frac{g}{L} & 0 \end{bmatrix}$, $B_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $B_1(x) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Матрицы критерия качества (2): $Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$,

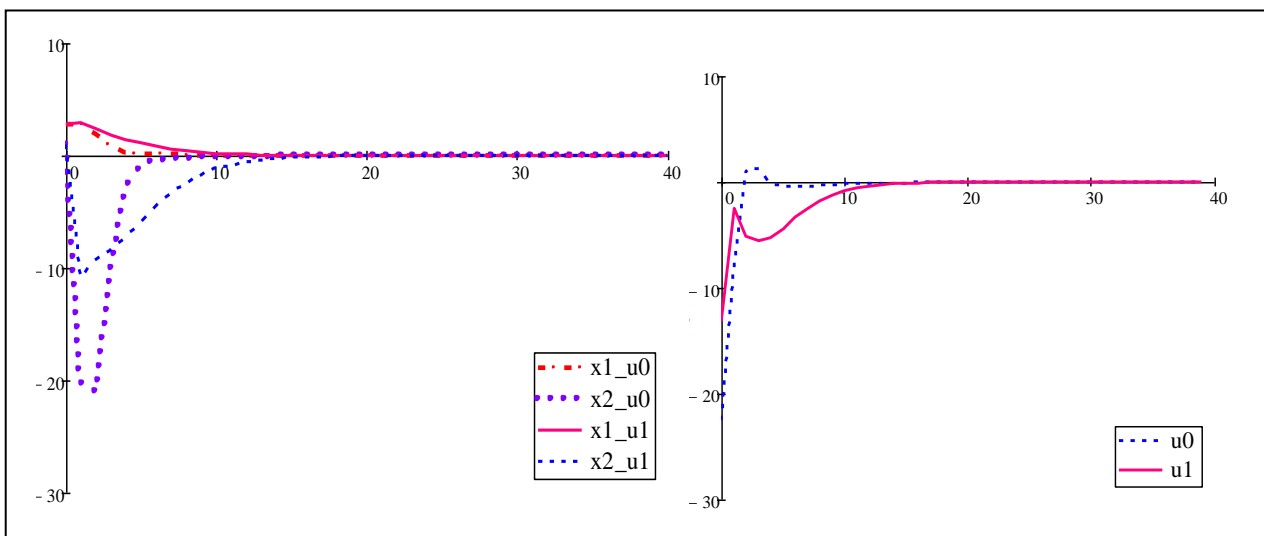


Рис. 1. –Траектории системы (20) при управлениях u_0 (линейный регулятор) и u_1 (нелинейный регулятор)

$Q_1(x) = \begin{bmatrix} 1500 + 0.1x_1^2 & 0 \\ 0 & 30 + 0.1x_2^2 \end{bmatrix}$, $R_0 = 1$. Траектории системы, полученные вдоль нелинейного регулятора (8) и, для сравнения, вдоль линейного регулятора $u_0(x) = K_0x$ при $x_0 = \begin{bmatrix} 2.8 \\ 1 \end{bmatrix}$ и числе шагов $N = 40$, представлены на Рис. 1.

Предположим, что в матрицах A_0, B_0 присутствует параметрическая неопределенность

$$A_0(\theta_1) = \begin{bmatrix} 1 & T_s \\ \frac{T_s g}{L} & 1 \end{bmatrix} + \theta_1 \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & -\frac{T\gamma}{L} \end{bmatrix}, \quad \theta_1 = \frac{1}{M}, M \in [0.05; 0.15], \theta_1 \in [6.7; 20], B_0 = \begin{bmatrix} 0 \\ \theta_2 \end{bmatrix},$$

$\theta_2 \in [0.8; 1.2]$. В данном случае согласно условию IV необходимо были построены четыре вершины и найдены четыре положительно определенные матрицы P_i .

Находим вершины так, как было описано выше:

$$A_1 = A_{cl,0}(\underline{\theta}_1, \underline{\theta}_2), A_2 = A_{cl,0}(\underline{\theta}_1, \bar{\theta}_2), \text{ и } A_3 = A_{cl,0}(\bar{\theta}_1, \underline{\theta}_2), A_4 = A_{cl,0}(\bar{\theta}_1, \bar{\theta}_2).$$

Эксперименты показали, что нелинейный срединный регулятор обеспечивает больший запас устойчивости (собственные числа матрицы замкнутой системы меньше, т.е. дальше от границы устойчивости) и лучшее приближение к положению равновесия за $N=40$ шагов, чем линейный. См. подробные результаты в табл. 1.

Таблица 1. Проведенные эксперименты для срединных регуляторов (12) и (13)

	Эксперимент 1	Эксперимент 2
Значения параметров	$\theta = [7.10; 0.98]$	$\theta = [19.99; 1.13]$
Собственные числа матрицы замкнутой системы с нелинейным регулятором (N=40)	[0.5606; 0.3399]	[0.70524; 0.2744]
Собственные числа матрицы замкнутой системы с линейным регулятором (N=40)	[0.7761; 0.3933]	[0.8168; 0.0741]
Состояние системы на шаге N=40 (с нелинейным регулятором)	[2.3312e-09; -2.0486e-08]	[6.0920e-06; -3.5914e-05]
Состояние системы на шаге N=40 (с линейным регулятором)	[-5.9263e-05; 0.0003]	[0.0003; -0.0012]

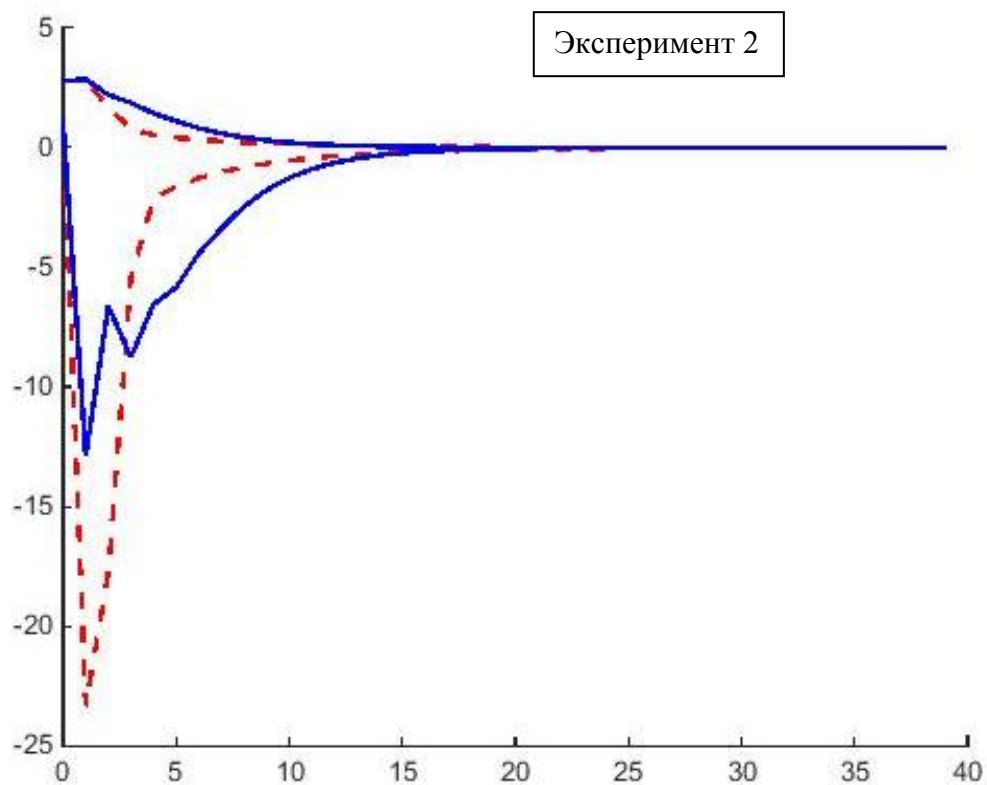
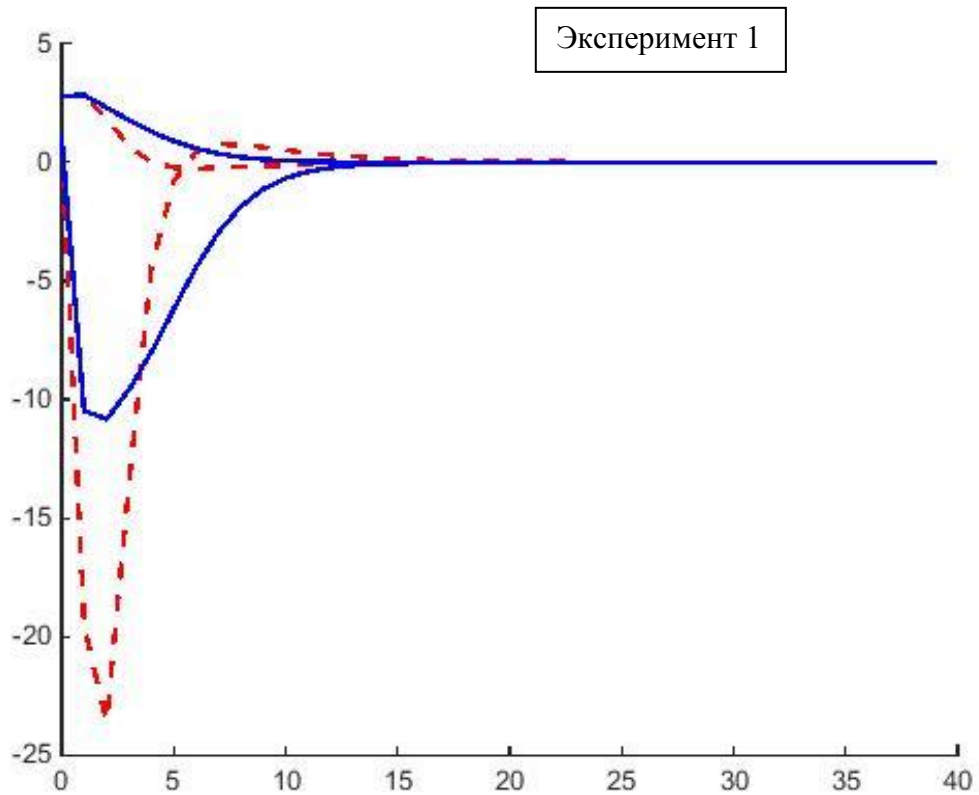


Рис. 2. –Траектории системы (20) при управлениях (12) (нелинейный срединный регулятор: сплошные линии) и (13) (линейный срединный регулятор: точка и тире) для экспериментов 1 и 2

При этом неравенство $\tilde{V}(x(t+1), \alpha) - \tilde{V}(x(t), \alpha) = x^T(t) \bar{D}(x, \mu, \theta, \alpha) x(t) < 0$ для функции Ляпунова (19), используемое для доказательства устойчивости по Ляпунову, при данном значении μ не выполняется. Тем не менее, робастность здесь сохраняется, так как условия теоремы только достаточные.

Предположим, в матрицу $A_1(x)$ вводятся детерминированные статические возмущения (ошибки) $A_1(x) = \begin{bmatrix} 0 & 0 \\ \frac{2g \sin(x_1)}{L \cdot x_1} - \frac{g}{L} & 0 \end{bmatrix} + \begin{bmatrix} 1.5x_1 & x_1 \\ x_1 & 1.5x_1 \end{bmatrix}$.

В этом случае эксперименты показали, что нелинейный срединный регулятор обеспечивает больший запас устойчивости (собственные числа матрицы замкнутой системы меньше, т.е. дальше от границы устойчивости) и лучшее приближение к положению равновесия за $N=40$ шагов, чем линейный. См. подробные результаты в табл. 2.

Таблица 2. Проведенные эксперименты для срединных регуляторов (12) и (13).

	Эксперимент 3	Эксперимент 4
Значения параметров	$\theta = [7.10; 0.98]$	$\theta = [19.99; 1.13]$
Собственные числа матрицы замкнутой системы с нелинейным регулятором	$[0,9554; 0,0634]$	$[0,9116; 0,4794]$
Собственные числа матрицы замкнутой системы с линейным регулятором	$[0,9987; 0,0860]$	$[0,9897; 0,3029]$
Состояние системы на шаге $N=40$ (с нелинейным регулятором)	$[-0,0574; 0,0491]$	$[0,0093; -0,0165]$
Состояние системы на шаге $N=40$ (с линейным регулятором)	$[-0,5644; -1,0627]$	$[-0,3731; -0,2097]$

Отметим, что неравенство $\tilde{V}(x(t+1), \alpha) - \tilde{V}(x(t), \alpha) = x^T(t) \bar{D}(x, \mu, \theta, \alpha) x(t) < 0$ для функции Ляпунова (19) выполняется уже с четвертого шага в третьем эксперименте, и с третьего шага для четвертого эксперимента, матрица $\bar{D}(x, \mu, \theta, \alpha)$ является отрицательно определенной, матрица $W(x, \mu, \theta)$ - положительно определенная, т.е. последние эксперименты демонстрируют робастность свойства асимптотической устойчивости замкнутой системы относительно интервальных неопределенностей в A_0, B_0 и детерминированных статических возмущений в $A_1(x)$.

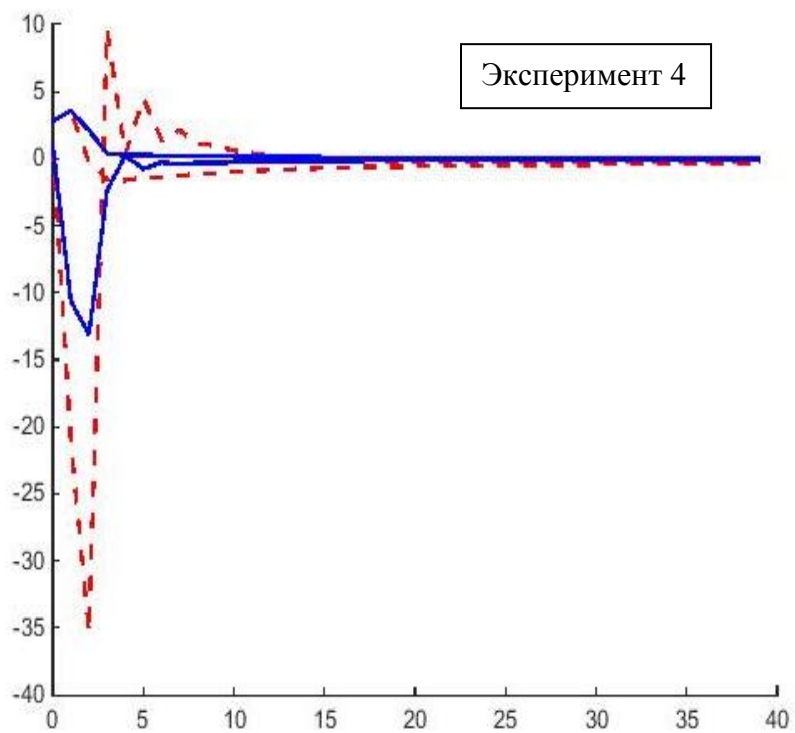
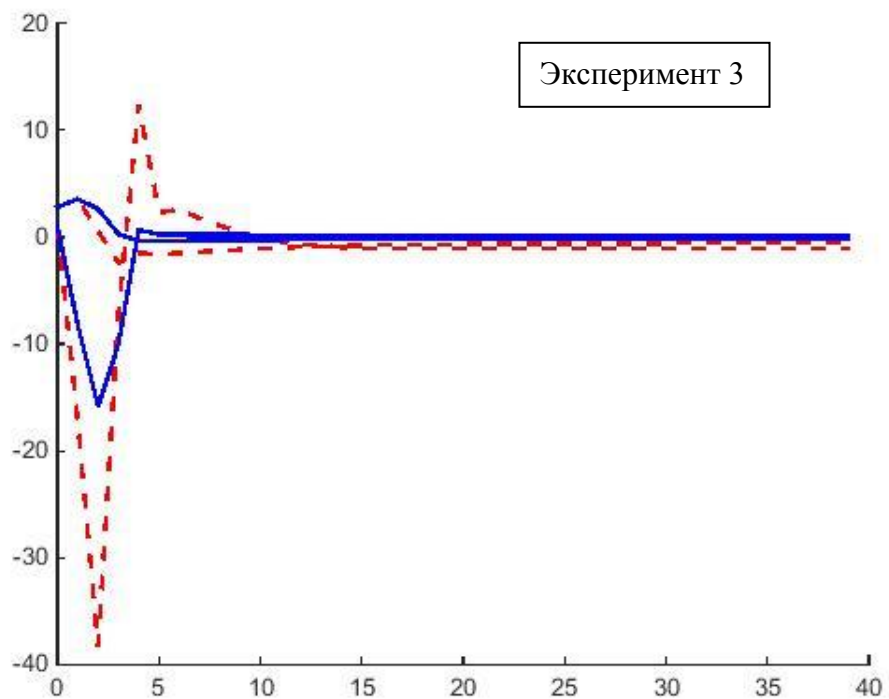


Рис. 3. –Траектории системы (20) при управлениях (12) (нелинейный срединный регулятор: сплошные линии) и (13) (линейный срединный регулятор: точка и тире) для экспериментов 3 и 4

4. Заключение

В данной работе приведено утверждение о робастности по отношению к параметрическим неопределенностям замкнутой дискретной квазилинейной системы вдоль нелинейного стабилизирующего «срединного» регулятора, построенного на основе дискретного уравнения Риккати с зависящими от состояния коэффициентами D-SDRE (слабой нелинейности), сформированное на основе разрешимости определенных линейных матричных неравенств. Приведенные эксперименты показали, что нелинейный срединный регулятор превосходит по своим характеристикам линейный срединный регулятор (оба рассчитаны при средних значениях параметров неопределенности). Таким образом, при устойчивости линейной стационарной системы при малых μ сохраняется устойчивость и нелинейной системы. Предлагаемый срединный нелинейный регулятор может быть использован для широкого класса дискретных систем управления с параметрическими неопределенностями и различными возмущениями.

Литература

1. Емельянов С.В., Даник Ю.Э., Дмитриев М.Г., Макаров Д.А. Стабилизация нелинейных дискретных динамических систем с параметром и с коэффициентами, зависящими от состояния// Доклады академии наук, том 466, №3. - 2016, С. 282–284.
2. Даник Ю.Э., Дмитриев М.Г., Макаров Д.А. Один алгоритм построения регуляторов для нелинейных систем с формальным малым параметром// Информационные технологии и вычислительные системы, №. 4, С. 35-44, 2015.
3. Oliveira M. C. De, Bernussou J., Geromel J. G. A new discrete-time robust stability condition// Systems and Control Letters, vol. 37, pp. 261-265, 1999.
4. Oliveira M. C. De, Geromel J. C., Hsu L. LMI characterization of structural and robust stability: the discrete-time case// Linear Algebra Appl., vol. 296, pp. 27-38, 1999.
5. Ramos D. C. W., Peres P. L. D. A less conservative LMI condition for the robust stability of discrete-time uncertain systems// System Control Lett., vol. 43, pp. 371-378, 2001.
6. Oliveira R.C.L.F., Peres P.L.D. Stability of polytopes of matrices via affine parameter-dependent Lyapunov functions: Asymptotically exact LMI conditions// Linear Algebra Appl., vol. 405(8), pp. 209–228, 2005.
7. Grman L., Rosinova D., Vesely V., Kozakova A.K. Robust stability conditions for polytopic systems// Internat. J. Systems Sci., vol. 36(15), pp. 961–973, 2005.
8. Oliveira R.C.L.F., P.L.D. Peres. Robust stability analysis and control design for time-varying discrete-time polytopic systems with bounded parameter variation// Proc. Amer. Control Conf., pp.3094 -3099, 2008.
9. Karimi A., Sadabadi M. S. Fixed-order controller design for state space polytopic systems by convex optimization// in 5th IFAC Symposium on System Structure and Control. Grenoble, France, 2013.
10. Kau S.-W., Liu Y.-S., Hong L., Lee C.-H., Fang C.-H., Lee L. A new LMI condition for robust stability of discrete-time uncertain systems// Syst. Control Lett., vol. 54, pp.1195 - 1203, 2005.

КОНЦЕПЦИЯ АРХИТЕКТУРЫ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА ДЛЯ ДИСТАНЦИОННОГО МОНИТОРИНГА АРКТИЧЕСКОЙ ЗОНЫ

Емельянова Ю.Г.

ИПС им. А.К. Айламазяна РАН
152021, Ярославская область, Переславский район, село Веськово, ул. Петра Первого,
дом 4 «а», yuliya.emelyanowa2015@yandex.ru

Хачумов В.М.

ИСА ФИЦ ИУ РАН
117312, Москва, просп. 60-летия Октября, д.9, vmh48@mail.ru

Аннотация

В работе рассмотрены структурная и функциональная организации автоматизированного рабочего места, предназначенного для операторов, осуществляющих мониторинг Арктической зоны Российской Федерации. Выделены основные направления его применения, включая задачи контроля Арктической зоны на основе дистанционного зондирования Земли, контроль и диагностику космических аппаратов арктического назначения, эколого-экономический мониторинг. Большое внимание уделено вопросам выбора инструментальных средств и технологий, необходимых для создания систем дистанционного мониторинга, а также способам удобного для восприятия отображения ситуаций в интерфейсе аналитика. Специальные исследования планируется провести в области когнитивного графического представления данных, способствующего их пониманию. Предполагается, что на основе проведенных исследований будет построена территориально распределенная сеть для мониторинга, анализа и принятия оперативных решений в случае возникновения чрезвычайных ситуаций.

1. Введение

Территорию Арктической зоны Российской Федерации (АЗРФ) следует разделять на группы: 1) сухопутные территории, 2) морские пространства, 3) исключительная экономическая зона и континентальный шельф, 4) воздушное пространство [1, 2]. К сожалению, точная граница Арктической зоны РФ не определена. В настоящее время к Арктической зоне РФ относятся внутренние воды и территориальное море, прилегающие к территориям субъектов Российской Федерации, землям и островам, расположенным в Северном Ледовитом океане, воздушное пространство над сухопутными территориями, морскими пространствами, над исключительной экономической зоной и континентальным шельфом АЗРФ [1]. В целях решения проблем АЗРФ, имеющих глобальный характер, требуется разработать новые методы и средства для комплексного решения задач дистанционного мониторинга Земли (ДЗЗ). Наиболее перспективным методом в настоящее время является мониторинг [3], включающий в регулярные наблюдения анализ получаемых сведений, оценку и прогнозирование состояния АЗРФ.

Предполагается, что автоматизированные рабочие места (АРМ) «Арктика» будут располагаться на станции приема, сбора, анализа и обработки данных, полученных от спутников ДЗЗ, наземных радиолокационных станций, станций ГЛОНАСС и летательных аппаратов, включая беспилотные. Сеть АРМ «Арктика» станет частью глобальной системы наблюдения АЗРФ, которая будет предоставлять комплексную информационную поддержку [4, 5], решать разнообразные прикладные и научные задачи ДЗЗ, обладать развитыми технологиями интерпретации и использования получаемых космиче-

ских данных [6]. Предлагаемые рабочие места аналитиков предназначены для скоростной обработки данных, высокоточного поиска информации, прогнозирования развития чрезвычайных ситуаций и поддержки принятия решений [7]. Основным назначением АРМ «Арктика» является интеллектуальная обработка полученных данных и выявление наиболее существенной целевой информации [8]. Задачи, которые будет решать АРМ «Арктика», можно разделить на группы по следующим трем типам: 1) контроль АЗРФ, 2) контроль космических аппаратов (КА) орбитальной группировки «Арктика» и других задействованных космических систем, 3) контроль эколого-экономического состояния.

2. Назначение АРМ «Арктика» по основным направлениям

2.1. Проблемы Арктической зоны РФ

В настоящее время актуальными проблемами для России являются обеспечение динамичного социально-экономического развития ее северных регионов, богатых природными ресурсами (в частности, углеводородными), решение задач по эффективной навигации по Северному морскому пути, а также по безопасному полету авиации в высоких широтах. Труднопредсказуемые аномальные явления природного и техногенного характера могут приводить к значительным людским и экономическим потерям. В этих условиях обостряется проблема повышения достоверности среднесрочных и долгосрочных прогнозов аномальных явлений в Арктической зоне РФ [5].

Велика роль Арктики и в обороноспособности России, в контроле над космическим пространством, в сдерживании военно-политического присутствия западных держав в северном геополитическом пространстве. Здесь дислоцирована значительная часть вооруженных сил России, в том числе большая часть Тихоокеанского флота – крупнейшие оперативно-стратегические объекты, являющиеся силовой основой решения задач национальной морской политики на соответствующих региональных направлениях [9].

Намеченные в рамках концепции исследования направлены на решение фундаментальной научной проблемы создания новых методов мониторинга Арктической зоны по данным дистанционного зондирования. В условиях роста потоков спутниковой информации (мультиспектральные и полноцветные снимки, данные телеметрии и др.) возрастает роль высокопроизводительной и, главное, интеллектуальной обработки для выявления наиболее существенной целевой информации. Исследования, таким образом, направлены на решение фундаментальной задачи создания аппаратно-программных средств высокопроизводительной интеллектуальной обработки информации, доставляемой средствами космического наблюдения. В настоящее время эта задача применительно к мониторингу Арктической зоны еще не решена.

2.2. Задачи контроля АЗРФ на основе ДЗЗ

Перечень решаемых на сегодняшний день задач ДЗЗ, расширенный с учетом возможностей космических комплексов РФ, включает [4-6]:

- 1) научное изучение АЗРФ;
- 2) получение фундаментальных знаний о Земле;
- 3) ледовую разведку океанологические исследования и наблюдение за подстилающей поверхностью в морских пространствах АЗРФ;
- 4) гидрометеорологическое наблюдение, контроль глобальных атмосферных изменений;
- 5) мониторинг чрезвычайных ситуаций;
- 6) развитие информационной инфраструктуры;
- 7) землепользование;

8) лесное хозяйство, обнаружение очагов возгорания лесных пожаров.

Перечислим виды информации, которые должна предоставлять российская спутниковая метеорологическая система для решения основных задач гидрометеобеспечения ВС РФ [6]:

- 1) многоспектральные снимки облачного, снежно-ледового и растительного покровов Земли;
- 2) многоспектральные снимки земной поверхности в видимой и инфракрасной областях спектра;
- 3) результаты радиолокационных наблюдений для ледовой разведки по трассам Северного морского пути и других высокоширотных районов Мирового океана;
- 4) результаты зондирования в микроволновой области для оценки температуры поверхности океана и скорости приводного ветра независимо от облачности, выявления зон интенсивных осадков;
- 5) результаты спектрометрических измерений уходящего излучения Земли в специально подобранных участках инфракрасной и микроволновой областей спектра с целью определения вертикальных профилей температуры и влажности нижней атмосферы и полного ее влагосодержания;
- 6) результаты радиометрических измерений в инфракрасной и микроволновой областях спектра для определения температуры земной поверхности и верхней границы облаков;
- 7) результаты сбора данных с наземной сети платформ сбора данных.

2.3. Задачи контроля и диагностики КА арктического назначения

Вторая группа задач направлена на совершенствование систем контроля и диагностики КА, способных прогнозировать нештатные ситуации (НШС), обнаруживать изменения в работе аппаратуры и аномалии в потоках телеметрических данных в режиме реального времени [10]. Космический аппарат состоит из нескольких составных частей [11]: целевой аппаратуры, которая обеспечивает выполнение стоящей перед КА миссии; систем энергообеспечения, терморегуляции, радиационной защиты, управления движением и ориентации; бортового радиокомплекса и др. Внедрение современных компьютеризированных контрольно-измерительных комплексов и перспективных интеллектуальных технологий мониторинга, новых датчиков-сенсоров, систем и приборов создает предпосылки [12] для расширения круга решаемых задач в области контроля и диагностики, способных повысить отказоустойчивость и продлить срок эксплуатации космической техники [13]. Использование технологий искусственного интеллекта позволяет увеличить достоверность и оперативность [12] решения задач диагностики, поддержания работоспособности и управления КА. На основе интеллектуальной обработки телеметрической информации (ТМИ) планируется решать следующие задачи [3, 8, 12, 14, 15]:

- 1) диагностика группы спутников и выявление неисправности отдельных спутников [13];
- 2) оценка технических характеристик КА;
- 3) определение параметров и прогноз движения КА;
- 4) контроль ориентации КА;
- 5) контроль маневров КА;
- 6) расчет временных ограничений для учета в подсистеме планирования полета;
- 7) отображение телеметрической информации;
- 8) визуализация в реальном масштабе времени зоны видимости КА;

- 9) контроль и диагностика бортового комплекса типовой и целевой аппаратуры;
- 10) контроль качества работы автоматической аппаратуры;
- 11) выбор оптимальных режимов работы бортовой аппаратуры (БА).

На основе анализа стандартов и патентов, необходимых для построения перспективных систем мониторинга подсистем космических аппаратов [16], выбран следующий перечень контролируемых и отображаемых данных: результаты контроля и диагностики; прогноз движения и ориентации КА, источник нарушения функционирования КА; числовое и когнитивное представление ТМИ и других параметров.

2.4. Задачи эколого-экономического мониторинга

Эта группа задач включает:

- 1) экологический мониторинг;
- 2) поиск полезных ископаемых и энергоносителей;
- 3) сельское хозяйство;
- 4) мониторинг природных ресурсов;
- 5) наблюдение прибрежных зон и океанов;
- 6) контроль деятельности человека.

Перечисленные задачи служат основой для наблюдения за деятельностью населения и построения эколого-экономических моделей Арктической зоны. Модели необходимы для проведения численных экспериментов на ЭВМ и прогнозирования состояния основных характеристик, представляющих интерес.

3. Инструменты для решения задач дистанционного мониторинга Арктической зоны

Рассмотрим инструменты и технологии, повышающие качество работы с информацией дистанционного мониторинга. Так как необходимо организовывать легко воспринимаемое визуально и удобное взаимодействие со сложным программным обеспечением и разнотипными данными больших объемов, то выработаны некоторые общие подходы, которые упрощают ориентирование пользователя в интерфейсе системы ДЗЗ. Во-первых, большинство разработчиков программного обеспечения придерживаются пути «снимок – карта – геоинформационная система» [17]. Во-вторых, был разработан базовый универсальный тип web-интерфейса, структура которого представлена в работах [18, 19]. На основе технологии GEOSMIS создано несколько десятков специализированных картографических web-интерфейсов в различных системах дистанционного мониторинга [20], например, интерфейс для информационной системы дистанционного мониторинга лесных пожаров [21]. Управление интерфейсом базового типа имеет блочную структуру. Каждому типу данных и характерным функциям управления соответствуют свои модули в интерфейсе. На уровне пользователя для каждого модуля и типа данных в интерфейсе созданы соответствующие вкладки работы с данными, позволяющие группировать однотипные данные и операции с данными. Модульная структура позволяет конфигурировать интерфейс для работы с нужными типами данных. Для работы с разновременными данными в базовом интерфейсе реализованы однотипные блоки, которые позволяют управлять различными по типам спутниковыми данными и результатами их обработки:

- 1) локализованными данными высокого пространственного разрешения;
- 2) данными, охватывающими значительные территории;

3) сериями временных и пространственных композитов различных продуктов обработки спутниковых данных.

В базовом интерфейсе предусмотрена возможность выборки различных типов слоев для проведения их совместного дательного анализа. Выбранные слои (сцены, сесансы и т.д.) собираются в отдельной вкладке, в которой с ними можно осуществлять требуемые операции.

В работе [22] представлен подход, позволяющий встраивать в системы ДЗЗ блоки, созданные на основе технологий Business intelligence (BI), которые обеспечивают возможность анализа получаемой информации. Основной особенностью данного подхода является автоматическое, потоковое структурирование информации, получаемой на основе спутниковых данных, и преобразование ее в вид, позволяющий использовать ее в онлайн-интерфейсах, обеспечивающих представление и анализ данных ДЗЗ. Особенности BI-технологий могут быть эффективно использованы в системах дистанционного мониторинга. В частности, они дают возможность строить динамичные отчеты. Примеры специализированных интерфейсов для систем дистанционного мониторинга, использующих BI-технологии, представлены в работах [21, 23].

Специальные системы и технологии, позволяющие оперировать с большими, постоянно пополняющимися архивами данных [20], были использованы при разработке систем ДЗЗ, представленных в работах [24, 25]. Для реализации указанных технологий в ИКИ РАН созданы специальные программные комплексы, краткое описание которых приведено в статье [20]. Эти технологии, в частности, применяются для создания систем, использующих активные архивы. Активными называют такие архивы, в которые интегрированы инструменты, дающие возможность не только найти и выбрать необходимые данные, но и провести сразу их обработку и анализ.

Задача организации эффективной работы с данными ДЗЗ Арктической зоны предполагает создание ряда программно-аппаратных решений, необходимых для организации сети АРМ, являющихся местами работы аналитиков [7]. Для реализации АРМ «Арктика» применяется специальное средство разработчика – комплекс инструментальных программных средств «ПС НСКиД» [26], созданный в ИПС им. А.К. Айламазяна РАН. ПС НСКиД – нейросетевая система контроля телеметрической информации, диагностики подсистем космических аппаратов, обработки космических снимков. Архитектура системы представлена на рисунке 1.

Основными функциональными блоками ППС ИНС являются:

- **Управляющая моделирующая среда (УМС).** Управляющая моделирующая среда состоит из вычислительного ядра и набора обрабатывающих модулей, наиболее важными среди которых являются модули нейронных сетей, модули специальной и типовой обработки.
- **База знаний и данных (БЗ/БД)** – содержит информацию о предметной области и обрабатываемую информацию. База знаний представляет собой набор описаний типов данных, модулей системы, схем решения задач; настроек искусственных нейронных сетей; многопроцессорной системы и др. База данных содержит актуальную информацию об объекте, т.е. данные телеметрии, снимки и командную информацию.
- **Интерфейс пользователя** позволяет формировать схемы решения задач на параллельном вычислителе в конвейерно-параллельном режиме.

Указанный программный комплекс выбран основным инструментом, так как он оказался достаточно эффективным средством при создании ряда прикладных систем:

- программа, предназначенная для спектрографической «закраски» полноцветных и мультиспектральных изображений (ПС "Emarking");
- программный комплекс автономного позиционирования беспилотного летательного аппарата на основе системы технического зрения (ПК «Навигатор-М»);
- интеллектуальная система для обнаружения и предотвращения сетевых атак на распределенные вычислительные системы (ИПС «Заслон»);
- модуль мониторинга аномальной сетевой активности на основе искусственных нейронных сетей (Эгида-НС);
- программный комплекс для визуально-блочного проектирования нейросетевых прикладных систем (ПК «Nervetic 1.0»).

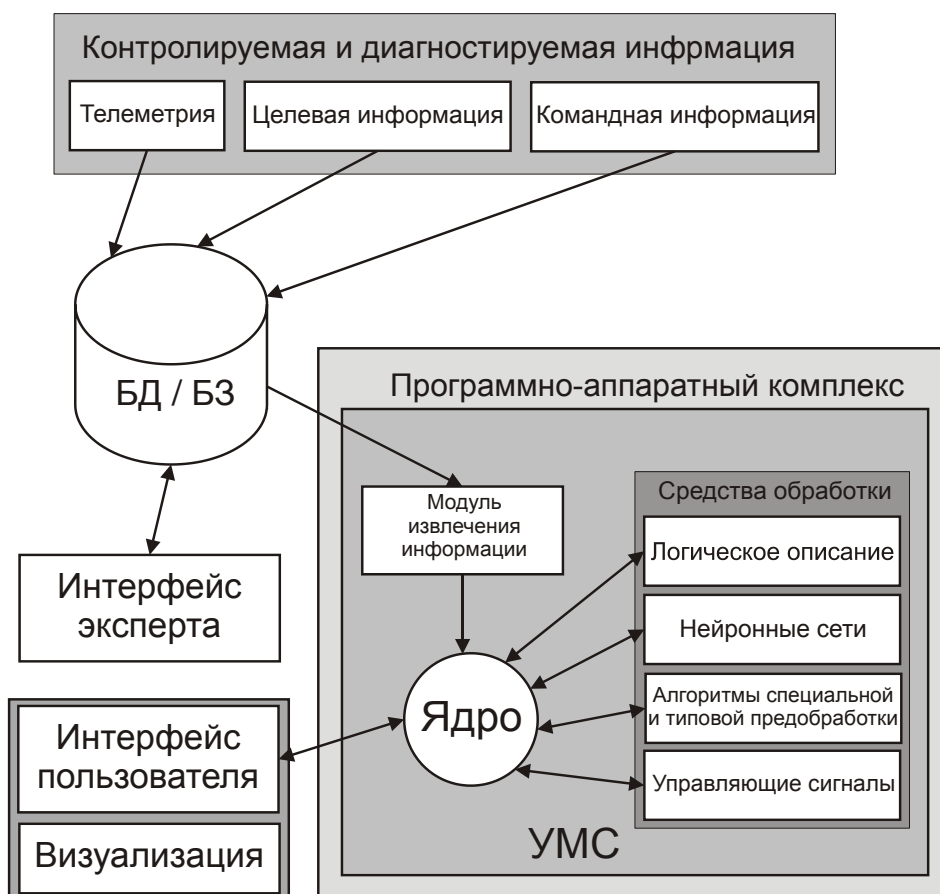


Рис. 1. – Архитектура нейросетевой системы контроля, диагностики и обработки изображений космического назначения

Сбор данных ДЗЗ осуществляется из открытых официальных источников. Например, интерфейс системы EarthExplorer позволяет просматривать и загружать изображения из множества EROS-архивов [27]. На сайте Научного Центра оперативного мониторинга Земли есть свободный доступ к небольшой выборке снимков, полученных от отечественных КА [28]. Кроме того, ведется сотрудничество с двумя организациями ракетно-космической промышленности, находящимися в ведении Роскосмоса: НИИ КС им. А.А. Максимова и АО «Российские космические системы». Технологическая схема предлагаемого к разработке АРМ представлена на рисунке 2.

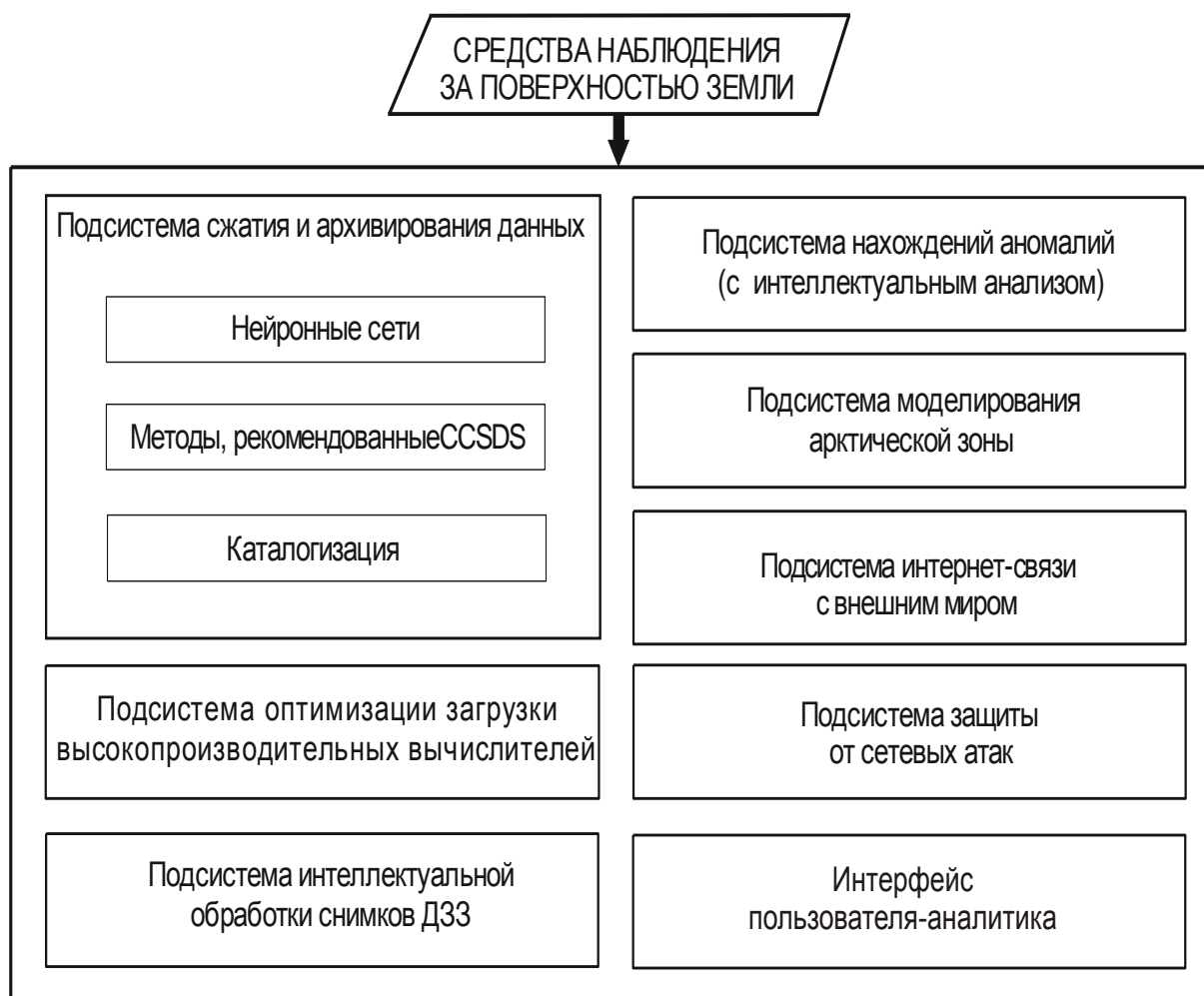


Рис. 2. – Технологическая схема АРМ «Арктика»

С использованием методов текстурного анализа предлагается решать задачи качественной кластеризации подстилающей поверхности и поиска целевых объектов [29]. Для распределения, обмена и компактного хранения данных используются разработанные в ИПС им. А.К. Айламазяна РАН нейросетевые средства сжатия и передачи данных как с потерями, так и без потерь.

4. Архитектура и функциональное назначение подсистем АРМ «Арктика»

Предполагается, что структура многоцелевой комплексной системы АРМ «Арктика» будет базироваться на типовой структуре космической системы ДЗЗ. Рассмотрим вариант структуры, адаптированный к условиям Арктической зоны (рис. 3.).

С целью совершенствования и развития перспективной системы освещения обстановки в АЗРФ, в ее состав будут включены новейшие космические средства, разработанные специально для Арктического региона, наземные радиолокационные станции и многофункциональные беспилотные комплексы. Уточненные функции АРМ «Арктика» представлены в табл.1-3.

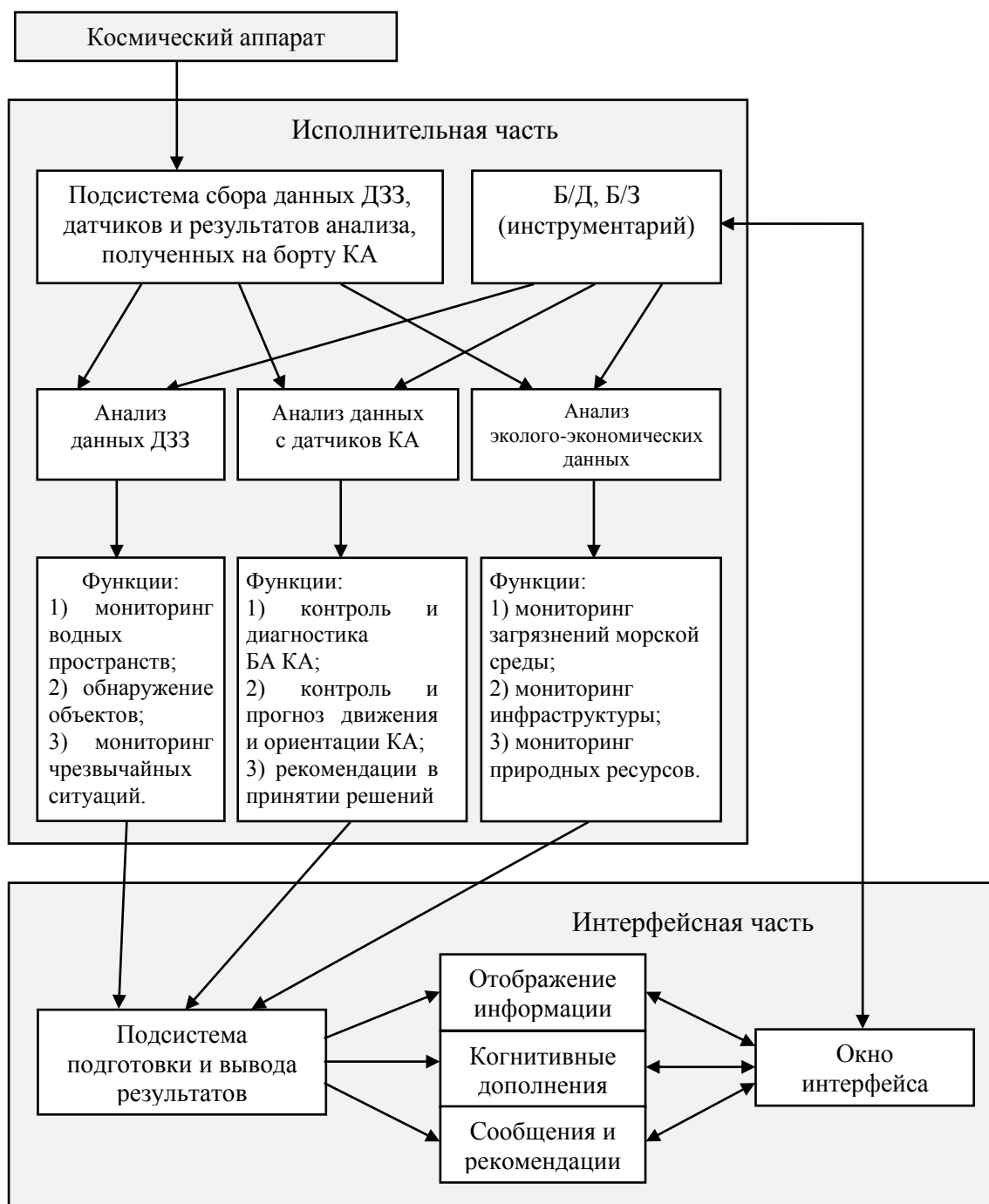


Рис. 3. – Функциональная схема АРМ «Арктика»

Таблица 1 – Функции АРМ «Арктика»: анализ данных ДЗЗ

№	Функции анализа	Подфункции		
1	Мониторинг водных пространств	Ледовая разведка		
		Наблюдение за подстилающей поверхностью в АЗРФ		
		Океанологические исследования		
		Наблюдение прибрежных зон и океанов		
		Контроль водных ресурсов		
		Мониторинг природных ресурсов		
		Обнаружение загрязнений		
		Обнаружение разливов нефти		
2	Обнаружение объектов	Водный транспорт. Цели обнаружения водного транспорта: 1) поиск и спасение морских и воздушных судов; 2) наблюдение и контроль за расположением и маршрутами иностранных судов; 3) обеспечение безопасности морского судоходства в Арктической зоне; 4) контроль трасс Северного морского пути с целью обеспечения безопасности мореплавания в этих районах; 5) регулирование движений судов и осуществление координации морских ледовых операций, установление районов обязательной ледокольной и лоцманской проводки всех судов и определение ее условий; 6) надзор за судами и другими плавучими средствами, которые могут быть источниками загрязнения; 7) обнаружение рыболовства вне рыболовных зон		
		Воздушный транспорт. Цели обнаружения воздушного транспорта: 1) идентификация объектов и принадлежность летательного аппарата; 2) обнаружение несанкционированных полетов.		
		Наземный транспорт. Цели обнаружения наземного транспорта: 1) наблюдение за транспортировкой нефти и газа; 2) выявление нарушений правил и ограничений на использование вездеходного оборудования в зоне тундры		
		Поиск полезных ископаемых и энергоносителей		
		Контроль деятельности человека с целью: 1) обнаружения незаконного строительства; 2) обнаружения незаконных установок по добыче природных ресурсов		
		Развитие информационной инфраструктуры		
		Мониторинг научных экспедиций		
		3	Мониторинг чрезвычайных ситуаций	Обнаружение очагов возгорания лесных пожаров
				Гидрометеорологическое наблюдение
				Контроль глобальных атмосферных изменений
Высокооперативный мониторинг землетрясений				
Обнаружение и прогнозирование криогенных опасных явлений				
Получение фундаментальных знаний о Земле				

Таблица 2 – Функции АРМ «Арктика»: Анализ данных с датчиков КА

№	Функции анализа	Подфункции
1	Контроль и диагностика бортовой аппаратуры	Диагностика группы спутников на высоком уровне, и выявление неисправности отдельных спутников на низком уровне
		Контроль и анализ основных подсистем бортовой аппаратуры КА: 1) система управления движением, ориентации и стабилизации; 2) система приема и передачи информации; 3) система электроснабжения; 4) система терморегулирования; 5) полезная нагрузка (научная аппаратура)
		Контроль качества работы автоматической аппаратуры
		Контроль режимов работы бортовой аппаратуры с целью выбора наиболее оптимальных
		Оценка технических характеристик КА (маневренность)
		Обнаружение повреждений
		Обнаружение источника нарушения функционирования КА
		Визуализация в реальном масштабе времени зоны видимости космических аппаратов ДЗЗ
		Числовое и когнитивное отображение ТМИ
		Выявление аномалий в данных
		Диагностика дефектов
		Выявление причин аномальной работы аппаратуры
		Контроль и диагностики датчиков положения КА
		Выявление и анализ проблем целевого функционирования КА
		Анализ готовности к выполнению программы полета
Контроль восстановления штатных функций бортовых систем		
2	Контроль и прогноз движения и ориентации КА	Байесовский подход
		Прогноз движения КА
		Прогноз ориентации КА
		Планирование полета с учетом временных ограничений на работу бортовой и наземной аппаратуры
		Контроль орбитальных параметров
		Выявление отклонений в показаниях параметров от прогноза
		Контроль ориентации КА
		Контроль параметров движения КА
		Расчет и контроль маневров КА
3	Рекомендации в принятии решений для передачи в бортовой / наземный контур управления	Необходимые корректирующие действия
		Перспективы по совершенствованию аппаратуры

Таблица 3 – Функции АРМ «Арктика»: Анализ эколого-экономических данных

№	Функции анализа	Подфункции
1	Мониторинг загрязнений морской среды	Предупреждение и обнаружение разлива нефти
		Оценка воздействия добычи нефти и газа на окружающую среду
		Предупреждение и обнаружение сброса радиоактивных отходов в морскую среду
2	Мониторинг инфраструктуры (так как ухудшение состояния вечной мерзлоты представляет опасность для инфраструктуры)	Мониторинг дорог
		Мониторинг нефть и газопроводов
		Мониторинг резервуаров для хранения
		Мониторинг нефтяных месторождений
		Мониторинг зданий
		Мониторинг строительства и функционирования крупных металлургических и иных промышленных комплексов
3	Мониторинг природных ресурсов	Наблюдение за разрушением береговой линии у Арктических морей
		Наблюдение и контроль за национальными парками и заповедниками
		Определение расположения и состояния колоний птиц и животных
		Наблюдение за прибрежной зоной (колонии птиц, растительность, береговая линия, солончаки)
		Наблюдение за «оазисами жизни» (полыньи) в покрытых льдом морях
		Наблюдение за миграцией животных с целью запрета низких полетов над колониями морских млекопитающих
		Выявление нарушений правил по посещению лежбищ, обнаружение полета над лежбищем морских млекопитающих
		Наблюдение за рыболовной зоной с целью обнаружения нелегального рыболовства
		Контроль за передвижением оленьего стада и состоянием пастбищ
		Контроль за изменениями в условиях обитания морских млекопитающих
		Контроль за изменениями и разрушениями растительного покрова

5. Анализ способов отображения ситуаций в Арктической зоне РФ

Для облегчения формирования задач ДЗЗ, в том числе поиска и распознавания целевых объектов, предлагается оснастить АРМ «Арктика» графическими интерфейсами с элементами когнитивной визуализации данных. Когнитивный подход предполагает использование образного представления результатов интеллектуального анализа данных ДЗЗ. Наиболее эффективна когнитивная графика для быстрой оценки ситуации в целом, так как она позволяет, не анализируя большого количества разнородной информации, сделать вывод о ситуации и спрогнозировать ее развитие [8]. Одним из примеров применения методов машинной графики в АРМ является визуализация в реальном масштабе времени зоны видимости КА ДЗЗ [8].

На рис. 4 представлена схема интерфейса АРМ «Арктика», предназначенного для отображения актуальной информации. Основой интерфейса для АРМ «Арктика» являются следующие модули: «Картографическая информация», «Когнитивная графика», «Статистика», «Оценка и рекомендации». Данные, которыми оперирует интерфейс, извлекаются из «Базы Данных и Базы Знаний» посредством «Элементов управления». После манипуляций, произведенных в интерфейсе, оператор может сохранить итоговый проект в «Базу Знаний».

Модуль «Картографической информации» предназначен для выполнения функций по работе со спутниковыми снимками и отображения результатов анализа контролируемых данных АЗРФ в графической форме.

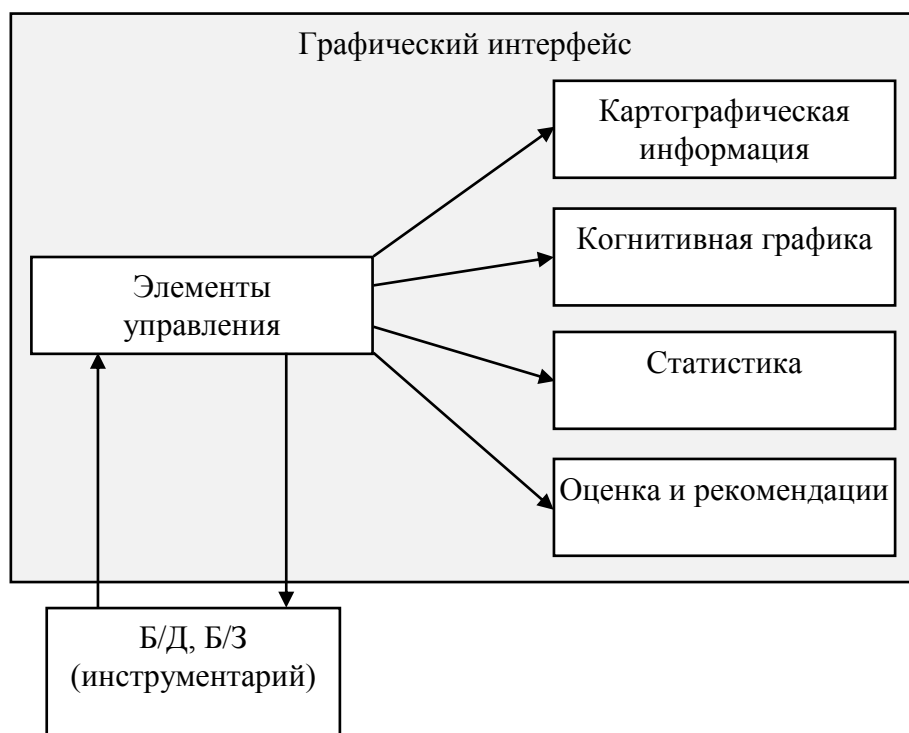


Рис. 4. – Функциональная схема интерфейса АРМ «Арктика»

Модуль «Когнитивная графика» обобщает в нескольких изображениях все контролируемые многомерные и многокритериальные данные. Когнитивный подход к отображению параметров позволяет формировать интуитивное представление о состоянии наблюдаемых объектов и систем, минуя трудоемкий анализ информации, представленной в статистическом виде (графики, диаграммы, таблицы, тексты и др.).

Модули «Статистика» и «Оценка и рекомендации» служат для просмотра статистических отчетов в виде графиков и таблиц, а также для отображения результатов анализа в текстовой форме.

6. Заключение

В работе представлена концепция автоматизированного рабочего места, предназначенного для выполнения ряда важнейших функций по мониторингу Арктической зоны РФ. Концепция опирается на анализ данных ДЗЗ, контроль и диагностику КА, эколого-экономический мониторинг. Дано развернутое представление о назначении каждой решаемой задачи. Составной частью АРМ «Арктика» является его графический интерфейс, на который возложены задачи оперативного и наглядного представления

анализируемых и обработанных данных. Предполагается, что сеть из подобных АРМ позволит аналитикам вести наблюдения за состоянием региона в целом и принимать оперативные решения.

Работа выполнена при финансовой поддержке проекта РФФИ №15-29-06945 офи_м «Развитие моделей, методов и программных средств обработки мультиспектральных снимков, видеопотоков и данных телеметрии для задач космического мониторинга Арктической зоны».

Литература

1. Проект федерального закона «Об Арктической зоне Российской Федерации». [Электронный ресурс]: Мурманский вестник: ежедневное областное издание. Дата обновления: 13.02.2013. URL: <http://www.mvestnik.ru/shwpgn.asp?pid=2013021311> (дата обращения: 15.03.2016).
2. Однопозов П. С. Международно-правовой режим арктических морских пространств / П. С. Однопозов // Правоведение, № 4, 1973, с. 78 – 82. Дата обновления: 28.06.2004. URL: <http://law.edu.ru/article/article.asp?articleID=1157296> (дата обращения: 15.03.2016).
3. Ежов С.А., Бурдаев М.Н., Кальмин А.В., Крылов И.А. Перспективы использования пилотируемых орбитальных комплексов для решения задач дистанционного зондирования Земли. – Информационно-измерительные и управляющие системы, №7, Т.5, 2007, с.23-25. URL: <https://sites.google.com/site/iamforintelligent/home/publications> (дата обращения: 15.03.2016).
4. Космическая система (КС) «Арктика». Цели создания и задачи КС «Арктика». [Электронный ресурс]: Новости Роскосмоса // Федеральное космическое агентство. Дата обновления: 13.08.2010. URL: <http://www.federalspace.ru/11669/> (дата обращения: 15.03.2016).
5. Асюшкин В.А., Грешилов П.А., Ефанов В.В., Мартынов М.Б., Моишеев А.А., Немыкин С.А., Романов В.М., Шевалев И.Л. Автоматические космические аппараты для фундаментальных и прикладных научных исследований / Под общ. ред. Полищука Г.М. и Пичхадзе К.М. М.: Изд-во МАИ-Принт, 2010. 659 с.
6. Карпенко А.В. Космические средства дистанционного зондирования земли (ДЗЗ). [Электронный ресурс]: Военно-технический сборник «Бастион»: журнал оборонно-промышленного комплекса. Дата обновления: 28.01.2016. URL: <http://bastion-karpenko.ru/space-means-remote-sensing/> (дата обращения: 15.03.2016).
7. Талалаев А.А., Фраленко В.П., Хачумов В.М., Шишкин О.Г. Сеть автоматизированных рабочих мест для высокопроизводительной обработки данных дистанционного зондирования арктической зоны. – Материалы IV Национального Суперкомпьютерного Форума (НСКФ-2015) (Переславль-Залесский, 24-27 ноября 2015 г.). – Переславль-Залесский: ИПС им. А.К. Айламазяна РАН, 2015. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfZzd3eVdvTVNDWms> (дата обращения: 15.03.2016).
8. Хачумов В.М. Методы когнитивной визуализации данных дистанционного зондирования Земли // Materials of the XI International scientific and practical conference “Proceedings of academic science”. – 2015. Vol.8 “Mathematics. Modern information technologies. Physics. Construction and architecture”. – Sheffield: Science and education Ltd., 2015, pp.31-34. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfdWFfd3RDLXBKbWM> (дата обращения: 15.03.2016).
9. Ковалев А. Международно-правовой режим Арктики и интересы России. – Индекс безопасности, Т.15, №3-4 (90-91), 2009, с.115-124. URL:

<http://www.pircenter.org/media/content/files/0/13407261070.pdf> (дата обращения: 15.03.2016).

10. Хачумов В.М. Нейросетевые технологии обработки информации в космических приложениях. – Труды Третьей Всероссийской научной конференции «Методы и средства обработки информации» (МСО-2009, 6-8 октября 2009 года). – М.: Издательский отдел факультета ВМиК МГУ имени М.В.Ломоносова; МАКС-Пресс, 2009, с.420-425. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfblVvVlZKenpqcW8> (дата обращения: 15.03.2016).
11. Гуцин В.Н. Основы устройства космических аппаратов: учебник для вузов. — М.: Машиностроение, 2003. — 272 с. — ISBN 5-217-01301-X.
12. Абрамов Н.С., Ардентов А.А., Емельянова Ю.Г., Талалаев А.А., Фраленко В.П., Шишкин О.Г. Архитектура системы мониторинга и прогнозирования состояния космического аппарата. – Программные системы: теория и приложения, Т.6, №2(25), 2015, с.85-99. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfWmdpTDZ1NURJd0E> (дата обращения: 15.03.2016).
13. Абрамов Н.С., Заднепровский В.Ф., Талалаев А.А., Фраленко В.П. Применение искусственных нейронных сетей в задачах контроля и диагностики подсистем космических аппаратов. – Современные проблемы науки и образования, №3, 2014. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfX3U4WHUtW9xY0E> (дата обращения: 15.03.2016).
14. Бурдаев М.Н. О контроле и коррекции внеатмосферного участка траектории спуска с орбиты искусственного спутника планеты. – Программные системы: теория и приложения, Т.5, №2(20), 2014, с.13-18. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfOGZTajhKZDhUREE> (дата обращения: 15.03.2016).
15. Бурдаев М.Н. О форме и размерах границ зон обзора поверхностей планет с космических аппаратов. – Пилотируемые полеты в космос, №12, 2014, с.71-75. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfY1p2eWpObTJjeE0> (дата обращения: 15.03.2016).
16. Талалаев А.А., Фраленко В.П., Хачумов В.М. Обзор стандартов и концепция построения средств мониторинга, контроля и диагностики космического аппарата. – Программные системы: теория и приложения, Т.6, №3(26), 2015, с.21-43. URL: <https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfVnQm9Hd2xQMek> (дата обращения: 15.03.2016).
17. Программный комплекс «Image Media Center». [Электронный ресурс]: Виртуальная лаборатория дистанционного обучения спутниковой гидрометеорологии // Всемирная метеорологическая организация РМУЦ ВМО в РФ. URL: http://meteovlab.meteorf.ru/index.php?option=com_content&view=article&id=792 (дата обращения: 15.03.2016).
18. Толпин В.А., Балашов И.В., Ефремов В.Ю., Лупян Е.А., Прошин А.А., Уваров И.А., Флитман Е.В. Создание интерфейсов для работы с данными современных систем дистанционного мониторинга (система GEOSMIS) // Современные проблемы дистанционного зондирования Земли из космоса. 2011. Т.8. №3. с.93-108. URL: <http://jr.rse.cosmos.ru/article.aspx?id=900> (дата обращения: 15.03.2016).
19. Толпин В. А. Система обработки и визуализации данных дистанционного зондирования Земли для мониторинга состояния сельскохозяйственной растительности: автореферат дис. ... кандидата технических наук: 05.13.01 / Толпин Владимир Аркадьевич; [Место защиты: Рязан. гос. радиотехн. акад.] – Рязань, 2013, 16 с.
20. Лупян Е.А., Балашов И.В., Бурцев М.А., Ефремов В.Ю., Кашницкий А.В., Кобец Д.А., Крашенинникова Ю.С., Мазуров А.А., Назиров Р.Р., Прошин А.А., Сычугов

- И.Г., Толпин В.А., Уваров И.А., Флитман Е.В. Создание технологий построения информационных систем дистанционного мониторинга // Современные проблемы дистанционного зондирования Земли из космоса. 2015. Т.12. №5. с.53–75. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1437> (дата обращения: 15.03.2016).
21. Лупян Е.А., Барталев С.А., Ершов Д.В., Котельников Р.В., Балашов И.В., Бурцев М.А., Егоров В.А., Ефремов В.Ю., Жарко В.О., Ковганко К.А., Колбудаев П.А., Крашенинникова Ю.С., Прошин А.А., Мазуров А.А., Уваров И.А., Стыщенко Ф.В., Сычугов И.Г., Флитман Е.В., Хвостиков С.А., Шуляк П.П. Организация работы со спутниковыми данными в информационной системе дистанционного мониторинга лесных пожаров Федерального агентства лесного хозяйства (ИСДМ-Рослесхоз) // Современные проблемы дистанционного зондирования Земли из космоса. 2015. Т.12. №5. с.222–250. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1435> (дата обращения: 15.03.2016).
 22. Д.А. Кобец, И.В. Балашов, И.Д. Данилов, Е.А. Лупян, И.Г. Сычугов, В.А. Толпин. Использование VI-технологий для создания инструментов для анализа данных спутникового мониторинга // Современные проблемы дистанционного зондирования Земли из космоса. 2015. Т.12. №4. с.17–27. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1424> (дата обращения: 15.03.2016).
 23. Барталев С.А., Ершов Д.В., Лупян Е.А., Толпин В.А. Возможности использования спутникового сервиса ВЕГА для решения различных задач мониторинга наземных экосистем // Современные проблемы дистанционного зондирования Земли из космоса. 2012. Т. 9. № 1. с.49-56. URL: <http://jr.rse.cosmos.ru/article.aspx?id=971> (дата обращения: 15.03.2016).
 24. Лупян Е.А., Матвеев А.А., Уваров И.А., Бочарова Т.Ю., Лаврова О.Ю., Митягина М.И. Спутниковый сервис See the Sea – инструмент для изучения процессов и явлений на поверхности океана // Современные проблемы дистанционного зондирования Земли из космоса. 2012. Т.9. №2. с.251–261. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1034> (дата обращения: 15.03.2016).
 25. Митягина М.И., Лаврова О.Ю., Бочарова Т.Ю.. Спутниковый мониторинг нефтяных загрязнений морской поверхности // Современные проблемы дистанционного зондирования Земли из космоса. 2015. Т.12. №5. с.130-149. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1426> (дата обращения: 15.03.2016). Хачумов В.М., Тищенко И.П., Талалаев А.А., Константинов К.А., Фраленко В.П., Емельянова Ю.Г. Нейросетевая система контроля телеметрической информации, диагностики подсистем космических аппаратов, обработки космических снимков (ПС НСКИД) // Свидетельство о государственной регистрации программы для ЭВМ № 2012613261, дата приоритета: 18.11.2011, дата регистрации: 06.04.2012.
 26. EarthExplorer. [Электронный ресурс]: USGS science for a changing world // U.S. Department of the Interior. Дата обновления: 03.30.2016. URL: <http://earthexplorer.usgs.gov/> (дата обращения: 18.04.2016).
 27. Научный центр оперативного мониторинга земли. [Электронный ресурс]: Новости Центра // Федеральное космическое агентство. Дата обновления: 15.04.2016. URL: http://www.ntsomz.ru/news/news_center (дата обращения: 18.04.2016).
 28. Фраленко В.П. Методы текстурного анализа изображений, обработка данных дистанционного зондирования Земли. – Программные системы: теория и приложения, Т.5, №4(22), 2014, с.19-39. URL: http://psta.psisiras.ru/read/psta2014_4_19-39.pdf (дата обращения: 15.03.2016).

О ПРИМЕНЕНИИ ВОЗМУЩЁННОЙ ФУНКЦИИ ЛЯПУНОВА К ИССЛЕДОВАНИЮ РОБАСТНОСТИ ДИНАМИЧЕСКИХ СИСТЕМ

Каледин М.Л.

НИУ ВШЭ

125319, Москва, Кочновский пр-д, 3, mlkaledin@edu.hse.ru

Макаров Д.А.

ООО «ТСА»

117312, Москва, пр-т 60-летия Октября, 9, makarov@isa.ru

Аннотация

Показано, что метод обобщённых (возмущённых) функций Ляпунова, предложенный М.М. Хапаевым, можно интерпретировать в терминах робастности динамических систем по отношению к специальным классам нелинейных воздействий. Класс возмущений выделяется с помощью условий на среднее скалярного произведения возмущений и градиента функции Ляпунова для невозмущённой системы.

1. Введение

В различных приложениях зачастую возникают вопросы стабилизации динамических систем (линейных и нелинейных), которые подвергаются различного рода воздействиям и при этом их решение сводится к проверке устойчивости замкнутых систем, а также определению области робастности стабилизирующих регуляторов по отношению к неопределённым возмущениям. Изучению такого рода вопросов посвящено множество публикаций (см. библиографию в [1-3]).

Распространёнными подходами для исследования устойчивости в теории управления являются второй метод Ляпунова и метод вектор-функций Ляпунова (см. [4]). Настоящая работа посвящена ещё одному мало используемому на данный момент методу исследования устойчивости нелинейных систем на основе обобщённых функций Ляпунова, предложенному в работах М.М. Хапаева [5-7]. Суть этого подхода заключается в выделении возмущений в исходной нелинейной задаче, построении функции Ляпунова, устанавливающей устойчивость нулевого положения равновесия для «возмущённой» системы и наконец, проводится исследование возмущений, которое зачастую помогает установить устойчивость по Ляпунову, неустойчивость или асимптотическую устойчивость положения равновесия в исходной «возмущённой» задаче.

Последнее можно использовать для проверки *робастности* применяемых на практике стабилизирующих регуляторов по отношению к неопределённым воздействиям заданного класса функций. Приведём некоторые теоремы метода обобщённых функций Ляпунова и проиллюстрируем их примерами.

2. Обобщение второго метода Ляпунова и исследование «нейтрального случая»: основные определения

Рассмотрим «невозмущённую» систему обыкновенных дифференциальных уравнений и систему с возмущениями соответственно

$$\frac{dx}{dt} = f(t, x), \tag{1}$$

$$\frac{dx}{dt} = f(t, x) + R(t, x), \quad (2)$$

где $\dim x = n$, f, R – в общем случае n -мерные вещественные вектор-функции, определённые в области

$$P = \{t \geq 0, \|x\| < H\}, H > 0.$$

Под векторной нормой понимается стандартная евклидова норма $\|\cdot\|_2$ (нижний индекс для краткости опускается). Пусть выполнены условия существования и единственности решения задачи Коши для обеих систем, а система **Ошибка! Источник ссылки не найден.** имеет стационарную точку $x = 0$.

Требуется исследовать на устойчивость возмущённую систему **Ошибка! Источник ссылки не найден.** по переменным x в окрестности точки $x = 0$. Если в **Ошибка! Источник ссылки не найден.** это решение устойчиво по Ляпунову («нейтральный случай»), то устойчивость **Ошибка! Источник ссылки не найден.** определяется свойствами функции $R(t, x)$ [6]. Возмущения $R(t, x)$ могут включать в себя параметр $\mu \in (0, \mu_0]$ (μ_0 не обязательно мало), тогда пишут $R(t, x, \mu)$.

Пусть для невозмущённой системы существует положительно определённая функция Ляпунова V_0 , допускающая бесконечно малый высший предел и полная производная которой $\dot{V}_0 \leq 0$.

Определение 1. Возмущённой функцией Ляпунова называется функция $V = V_0 + u(t, x, \mu, \epsilon)$, где $u(t, x, \mu, \epsilon)$ может быть сколь угодно малым при достаточно малом μ .

Пусть $\phi(t, x, \mu, \epsilon) = (\text{grad } V, R(t, x))$, $x = \bar{x}(t)$ – решение невозмущённой задачи Коши **Ошибка! Источник ссылки не найден.** с начальным условием $x(t_0) = x_0$. Рассмотрим интеграл $J(t_0, x_0, T, \mu, \epsilon) = \int_{t_0}^{t_0+T} \phi(s, \bar{x}, \mu, \epsilon) ds$ или интегральное среднее от ϕ (если они существуют), определяемое как $\psi(t_0, x_0, \mu, \epsilon) = \lim_{T \rightarrow \infty} \frac{1}{T} J(t_0, x_0, T, \mu, \epsilon)$. Оказывается, что при определённых условиях, описанных в теоремах Хапаева (см. гл. 2 в [6]), отрицательность этих величин может заменить условие знакоотрицательности полной производной функции Ляпунова, возникающее в теоремах второго метода Ляпунова для доказательства асимптотической устойчивости.

3. Асимптотическая устойчивость нелинейных систем в нейтральном случае

Рассматриваются возмущённая и невозмущённая системы нелинейных уравнений (1)-(2). Последняя имеет устойчивое по Ляпунову положение равновесия $x = 0$. Для таких систем справедлива

Теорема 1 (с.100 в [7]). Пусть $t \geq 0$, $\epsilon_0 > 0$ и выполнены условия

- 1) $f, R \in \text{Lip}_x(L)$ и $\exists N, r: \|R\| \leq N\|x\|^r$, $N > 0, r > 1$.
- 2) $\exists V_0(t, x)$, допускающая бесконечно малый высший предел и, принимая во внимание невозмущённую систему, $\dot{V}_0 \leq 0$.

3) $\phi(t, x) = (\text{grad } V_0, R(t, x))$ дифференцируема по x и

$$\exists M > 0, d > r: |\phi| \leq M\|x\|^d, \|\nabla\phi\| \leq M\|x\|^{d-1}.$$

4) Существует среднее от ϕ , определённое как

$$\psi(t_0, x_0) = \lim_{T \rightarrow \infty} \left[\frac{1}{T} \int_{t_0}^{t_0+T} \phi(s, \bar{x}(s, t_0, x_0)) ds \right], \text{ и}$$

$$\forall \delta_1 \exists \delta_2 > 0: \psi(t_0, x_0) \leq -\delta_2 \text{ при } \|x\| > \delta_1.$$

5) $\|x\|^d / \psi(t, x) = O(1)$ при $\|x\| \rightarrow 0$.

6) $\exists l > 0: \forall t_0 \geq 0, \|x_0\| \leq \epsilon_0, t - t_0 \geq l$, что имеет место оценка

$$|\kappa(t, t_0, x_0)| \leq \frac{|\psi(t_0, x_0)|}{4}, \text{ где } \kappa(t, t_0, x_0) = \frac{\int_{t_0}^t \phi(s, \bar{x}(s, t_0, x_0)) ds}{t - t_0} - \psi(t_0, x_0).$$

Тогда $x = 0$ асимптотически устойчиво по Ляпунову.

Рассмотрим в качестве примера модель физического маятника, которая описывается системой обыкновенных уравнений второго порядка

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -c^2 \sin x_1 - r x_2, \end{aligned}$$

где c^2 – некоторая константа, включающая в себя физические параметры системы, $r > 0$ – постоянная силы трения, $x = [x_1 \ x_2]^T$. Используем Теорему 1) для доказательства асимптотической устойчивости $x_1 = 0, x_2 = 0$.

Для этого перепишем систему в виде возмущённой:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -c^2 x_1 + R(x_1, x_2) \\ R(x_1, x_2) &= +c^2 x_1 - c^2 \sin x_1 - r x_2. \end{aligned}$$

Вполне очевидно выполнение условий 1), функция $V = V_0$ – функция Ляпунова для линейного осциллятора, удовлетворяющая 2): $V_0 = \frac{c^2 x_1^2}{2} + \frac{x_2^2}{2}$. Рассмотрим теперь $\phi = x_2(c^2(x_1 - \sin(x_1)) - r x_2)$. Учитывая $|x_2| |x_1| \leq \frac{x_2^2 + x_1^2}{2}$, имеем

$$|\phi| \leq |x_2(c^2|x_1| - r x_2)| \leq c^2|x_2||x_1| + r x_2^2 \leq D_1\|x\|^2.$$

Оценим градиент ϕ :

$$\|\nabla\phi\| \leq \sqrt{4c^4 x_2^2 + (c^2|x_1| - 2r x_2)^2} \leq D_2\|x\|.$$

Можно выбрать наибольшее из D_1, D_2 , чтобы получить оценку в условии 3). Наибольшую сложность представляет проверка 4), можно в явном виде посчитать среднее (c_1, c_2 – произвольные константы в решении невозмущённой системы): $\psi(t_0, x_0) = -r \cdot (c_1^2 + c_2^2)/2$. Тогда условие 5) тоже верно, так как c_1, c_2 линейно выражаются через начальные условия x_0 , и $\|x\|^2 / \psi(t, x) = O(1)$ при $\|x\| \rightarrow 0$. Среднее $\psi(t_0, x_0)$ существует и, следовательно, имеет место оценка (6). Таким образом, возмущённая система имеет асимптотически устойчивое нулевое положение равновесия, что подтверждает численное моделирование системы (рисунки 1 и 2).

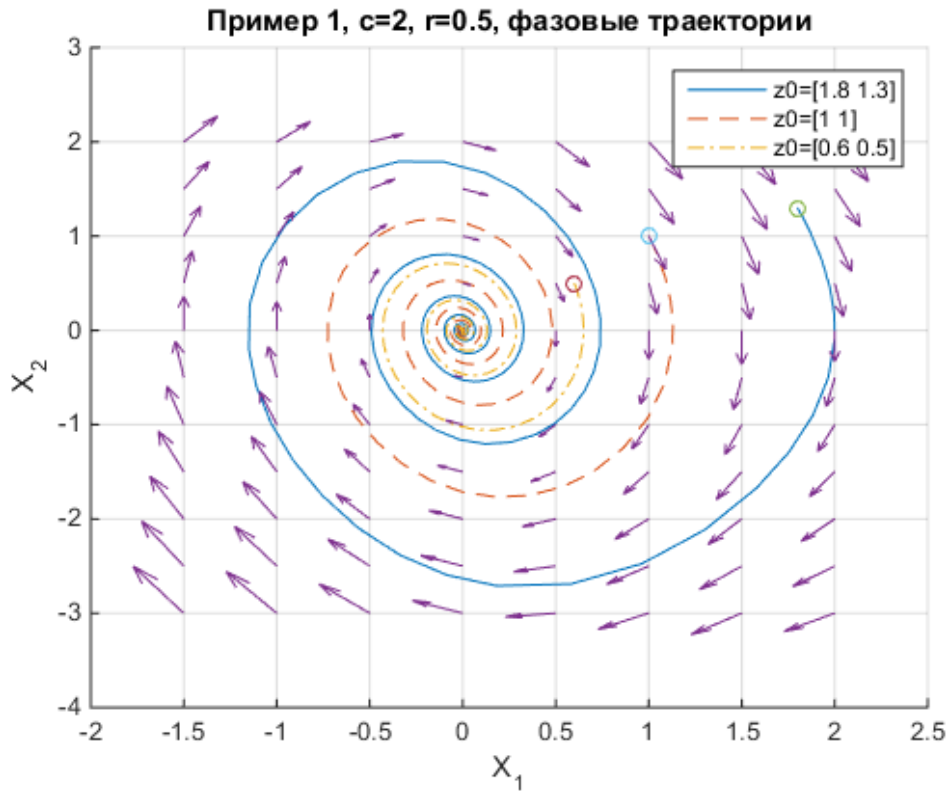


Рис. 1. Фазовые траектории нескольких решений (кружками обведены начальные условия)

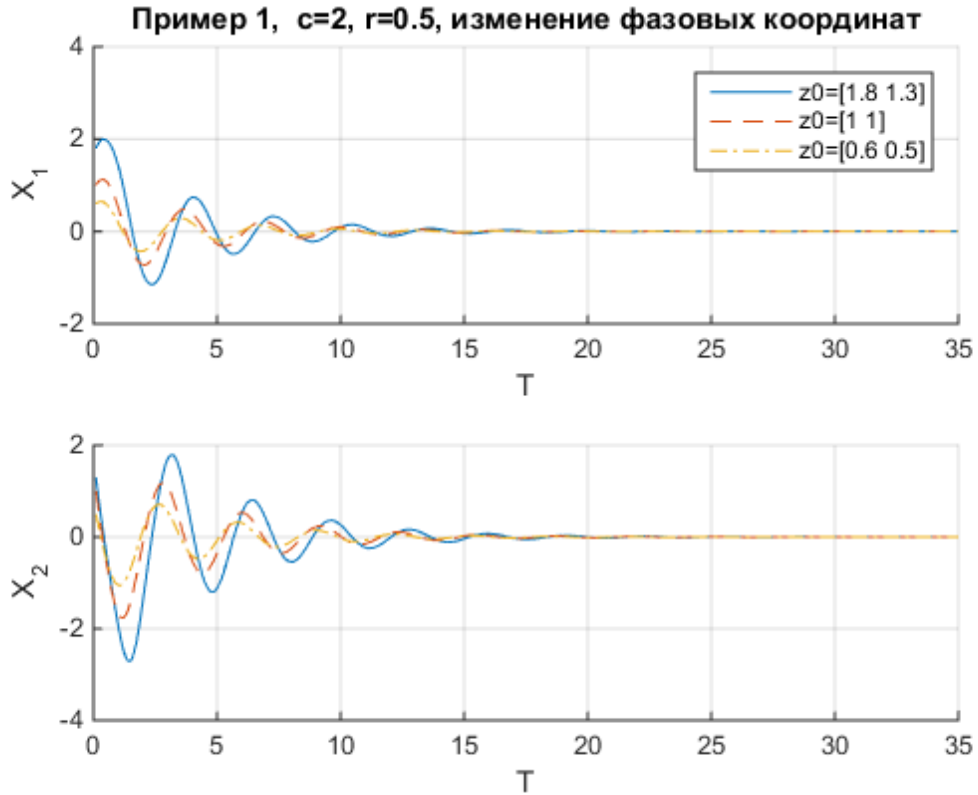


Рис. 2. Графики изменения фазовых переменных

4. Теорема об асимптотической устойчивости нелинейных систем с малыми возмущениями

Рассматривается возмущённая система нелинейных уравнений

$$\dot{x} = f(t, x) + \mu R(t, x),$$

с параметром $\mu > 0$. При этом невозмущённая система имеет устойчивую по Ляпунову точку $x = 0$. Наличие малого параметра позволяет использовать более слабые ограничения. Вместо условия 3) предыдущей теоремы для функции ϕ накладываются ограничения лишь на функцию Ляпунова невозмущённой системы. Для таких систем верна

Теорема 2 (стр.102 в [7]). Пусть для $t \geq 0$, $\epsilon_0 > 0$, $0 < \mu < \mu_0$ выполняются условия

- 1) $f, R \in Lip_x(L)$ и $\exists N, r: \|R\| \leq N\|x\|^r$, $N > 0, r \geq 1$.
- 2) $\exists V_0(t, x) \in C^2(x)$, допускающая бесконечно малый высший предел и, принимая во внимание невозмущённую систему, $\dot{V}_0 \leq 0$.
- 3) $\exists M_V > 0, m \geq 1: \|\text{grad } V_0\| \leq M_V \|x\|^m, \left\| \frac{\partial^2 V_0}{\partial x^2} \right\| \leq M_V \|x\|^{m-1}$.
- 4) Существует среднее от $\phi: \forall \delta_1 \exists \delta_2 > 0: \psi(t_0, x_0) \leq -\delta_2$ при $\|x\| > \delta_1$.
- 5) $\frac{\|x\|^{m+2}}{\psi(t, x)} = O(1)$ при $\|x\| \rightarrow 0$.
- 6) $\exists l > 0: \forall t_0 \geq 0, \|x_0\| \leq \epsilon_0, t - t_0 \geq l$ и имеет место $|x(t, t_0, x_0)| \leq \frac{|\psi(t_0, x_0)|}{4}$.

Тогда $x = 0$ асимптотически устойчиво.

Рассмотрим нелинейный осциллятор с трением, задающийся системой

$$\begin{aligned} \dot{x} &= v, \\ \dot{v} &= -\mu(1 + \sin x)v - x. \end{aligned}$$

Условия 1) очевидно выполняются, функция $V = \frac{x^2}{2} + \frac{v^2}{2}$ удовлетворяет 2). Так как это квадратичная форма, условие 3) выполняется с константой $m = 1$.

Среднее существует, оно ненулевое и отрицательное, так как решение $x(t), v(t)$ невозмущённого уравнения – ограниченные на бесконечности функции:

$$\forall \delta_1 > 0 \exists \delta_2 > 0: \|x\| < \delta_2 \Rightarrow \psi(t_0, v_0, x_0) = \lim_{T \rightarrow \infty} - \left[\int_{t_0}^{t_0+T} \mu(1 + \sin x)v(s)^2 ds \right] / T < -\delta_2.$$

Условие 5) выполнено при $\|x\| \rightarrow 0$: константы c_1, c_2 линейно выражаются через начальные условия. Отсюда следует $\|x\|^3 / \psi(t, x) \rightarrow const$ при $\|x\| \rightarrow 0$. В силу существования и конечности среднего 6) выполнено. Итак, возмущённая система асимптотически устойчива, что подтверждено численными экспериментами (рисунок 3).

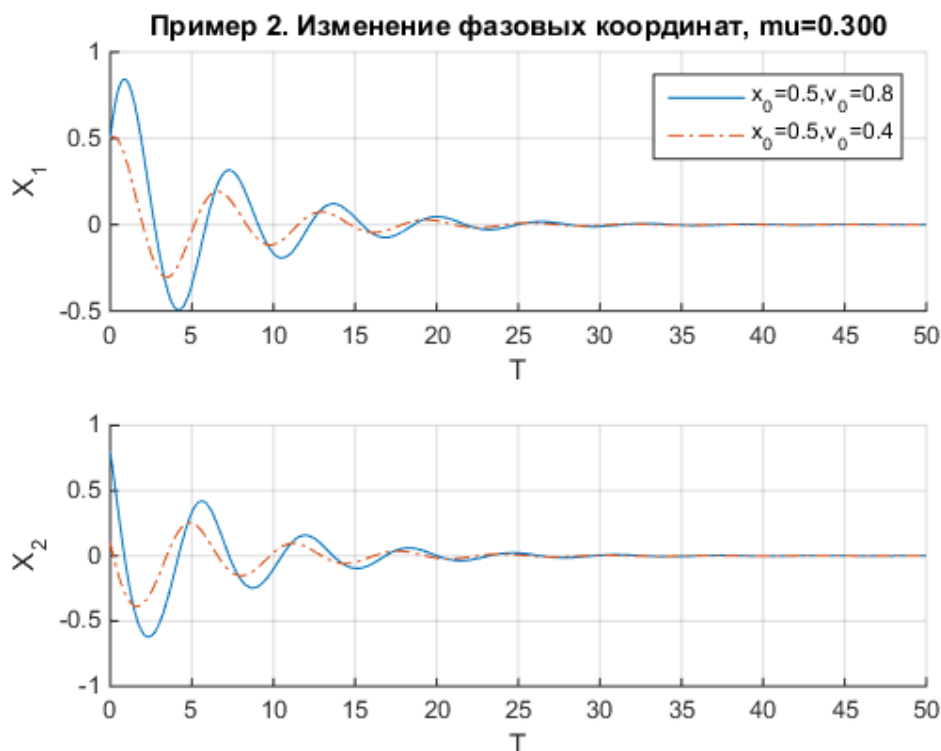


Рис. 3. Графики изменения фазовых переменных

5. Применение метода к исследованию робастности построенного управления

Введем

Определение 2. Классом Σ_μ для невозмущённой системы с функцией Ляпунова V_0 , обеспечивающей устойчивость нулевого решения, назовем класс воздействий $R(t, x)$, удовлетворяющих условиям 1), 3)-6) теоремы 2 с формально малым параметром $\mu > 0$.

Рассмотрим слабоуправляемую систему

$$\frac{dx}{dt} = f(t, x) + \mu u(t, x),$$

где $u(t, z)$ – управление. Пусть незамкнутая система

$$\frac{dx}{dt} = f(t, x)$$

устойчива, что обеспечивается существованием положительно определённой функции Ляпунова V_0 , допускающей беконечно малый высший предел, и $\dot{V}_0 \leq 0$. В этом случае можно искать малое стабилизирующее управление в классе Σ_μ . Особенностью такого подхода к стабилизации является обеспечение асимптотической устойчивости замкнутой системы с помощью малого расхода управления. Более того, само управление может быть нестационарным, что существенно расширяет область возможного применения в технических системах.

Определение 3. Классом Σ для невозмущённой системы с функцией Ляпунова V_0 , обеспечивающей устойчивость нулевого решения, назовем класс воздействий $R(t, x)$, удовлетворяющих условиям 1), 3)-б) на возмущения и среднее из теоремы 1.

Пусть имеется замкнутая система с построенным управлением $u = w(t, x)$

$$\frac{dx}{dt} = f(t, x) + \mu w(t, x)$$

и для неё построена функция Ляпунова V_0 , обеспечивающая, как минимум, устойчивость. В реальных системах часто возникают неопределённые возмущения, например, флуктуация параметров вследствие изменения режимов работы, выгорания топлива и т.д.. Кроме того, само управление $\mu w(t, x)$ может содержать шумы. Поэтому полная система выглядит так

$$\frac{dx}{dt} = f(t, x) + \mu w(t, x) + R(t, x),$$

где $R(t, x)$ – слагаемое, учитывающее возмущения различного рода. Если $R(t, x) \in \Sigma$ или $R(t, x) \in \Sigma_\mu$, то полная система будет асимптотически устойчивой. В этом случае говорят, что система робастна по отношению к возмущениям класса Σ и Σ_μ .

6. Заключение

Метод обобщённых функций Ляпунова может найти широкое применение в построении стабилизирующих регуляторов и проверке робастности управления по отношению к возмущениям определенных типов. Важным практическим аспектом подхода М.М. Хапаева к управляемым системам является обеспечение асимптотической устойчивости при управлении устойчивой нелинейной системой с помощью специальных типов возмущений, например, осциллирующего вида, определенных малых нестационарных управлений. Такой подход может найти применение, в частности, в области стабилизации летательных аппаратов, когда нецелесообразно применять большой расход управления.

Работа выполнена при финансовой поддержке проекта РФФИ № 16-38-60198 мол_a_дк.

Литература

1. Поляк Б.Т. Робастная устойчивость и управление / Б.Т. Поляк, П.С. Щербаков. – М.: Наука, 2002. 303 с.
2. Афанасьев В.Н. Управление неопределёнными динамическими объектами. – М.: Физматлит, 2008. 208 с.
3. Грязина Е.Н. Современное состояние метода D-разбиения / Е.Н. Грязина, Б.Т. Поляк, А.А. Трёмба // Автоматика и телемеханика. 2008. №12. С. 3-40.
4. Афанасьев В.Н. Математическая теория конструирования систем управления / В.Н. Афанасьев, В.Б. Колмановский, В.Р. Носов. – М.: Высшая школа, 2003. 614 с.
5. Хапаев М.М. Обобщение второго метода Ляпунова и исследование на устойчивость некоторых резонансных задач. // ДАН СССР. 1970. Т. 193. № 1. С. 46-49.
6. Хапаев М.М. Усреднение в теории устойчивости: Исследование резонансных многочастотных систем. – М.: Наука, 1986. 192 с.
7. Хапаев М.М. Асимптотические методы и устойчивость в теории нелинейных колебаний. – М.: Высшая школа, 1988. 184 с.

О РАЗРАБОТКЕ НЕЙРОСЕТЕВОГО НАСТРОЙЩИКА ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА КОНТУРА ТОКА ПРИ УПРАВЛЕНИИ ЭЛЕКТРОПРИВОДОМ ПРОКАТНОЙ КЛЕТКИ

Петров Владислав Анатольевич

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВО
Национальный исследовательский технологический университет "МИСиС"
309516, Белгородская обл., г. Старый Оскол, м-н Макаренко, д.42, 79040882508@ya.ru

Глуценко Антон Игоревич

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВО
Национальный исследовательский технологический университет "МИСиС"
309516, Белгородская обл., г. Старый Оскол, м-н Макаренко, д.42, strondutt@mail.ru

Еременко Юрий Иванович

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВО
Национальный исследовательский технологический университет "МИСиС"
309516, Белгородская обл., г. Старый Оскол, м-н Макаренко, д.42, erem49@mail.ru

Аннотация

В статье рассмотрена проблема повышения эффективности управления прокатной клетью, что в конечном итоге должно привести к снижению ее энергопотребления и, как следствие, принести экономический эффект. Данная задача может быть решена путем построения адаптивной системы управления такой клетью на основе оперативной настройки параметров линейных регуляторов, используемых во всех контурах управления объектом. В рамках исследования разработан нейросетевой настройщик параметров ПИ-регулятора контура тока, позволяющий подстраивать значения K_p и K_i в условиях изменения параметров обмотки якоря. Кроме того, если начальные значения данных коэффициентов не обеспечивают требуемое качество переходных процессов для постоянных параметров обмотки якоря, то система также произведет их настройку. В статье приведены функциональная схема настройщика, структура нейронной сети и база правил, определяющая моменты обучения такой сети и скорости обучения нейронов выходного слоя. Результаты проведенного численного эксперимента подтверждают эффективность предлагаемого подхода: при использовании настройщика удалось сократить энергопотребление клетки на 3-4%.

1. Введение

Металлургия является одной из основных отраслей промышленности в России. Одним из основных направлений развития металлургической промышленности, для поддержания конкурентоспособности производимого металла, в настоящее время является снижение затрат на его изготовление [1].

Среди основных потребителей энергоресурсов в металлургической отрасли необходимо выделить прокатное производство [2-4], включающее в свой состав прокатные станы. В свою очередь, главным потребителем электроэнергии в составе прокатного стана являются электропривода больших мощностей, используемые для вращения валков прокатных клеток. Основной задачей прокатной клетки является поддержание требуемого качества проката: соответствие геометрических размеров, формы, состояния поверхности прокатываемой заготовки. Снижение энергоемкости таких приводов даже на несколько процентов позволит существенно уменьшить

себестоимость производимой продукции. Наиболее энергоемкими прокатными клетями являются черновые клетки, первые в технологической цепочке стана.

Ключевое значение в решении данной задачи имеют вопросы автоматизации. В системах автоматизированного управления рассматриваемыми процессами, как правило, используются хорошо зарекомендовавшие себя П-, ПИ- и ПИД алгоритмы, как и в 90-95% контуров управления в промышленности в целом [5]. Популярность ПИД-регуляторов можно объяснить простотой их структуры, легкостью для понимания инженерным персоналом, и тем, что они позволяют достигать необходимого качества управления для большей части технологических объектов.

Кроме перечисленных достоинств применение П- и ПИ- регуляторов вносит требование к линейности ОУ, что и является основным недостатком подобной схемы для управления прокатной клетью, поскольку она, как объект управления, обладает существенными нелинейностями, т.е. в процессе работы клетки ее параметры изменяются. Это происходит по следующим причинам: характеристики приводного двигателя нелинейны, изменяются температура и геометрические размеры прокатываемой заготовки; изменяются скорости процесса прокатки; происходит износ оборудования, который вызывает неконтролируемое появление люфтов в механической части электропривода. Износ прокатного оборудования очень сложно контролировать. Из-за ударного изменения нагрузки при захвате и разгона привода при прокатке с заготовкой [6] момент нагрузки главного привода может увеличиваться в 1.5-2 раза при наличии зазоров [7]. Люфты в механической части привода увеличиваются от старения деталей. Это характерно для всех прокатных клеток, но выражено преимущественно в работе черновых клеток.

Учитывая описанные нелинейности можно сделать вывод о том, что для управления прокатной клетью недостаточно простого применения ПИД-алгоритма управления, особенно если параметры регулятора подбирать один раз при пуске системы и не производить их коррекцию в дальнейшем. В работе [8] указано, что в таких условиях ПИД законы в большинстве случаев не позволяют обеспечить требуемого качества регулирования во всех режимах работы клетки.

Данная проблема может быть решена путем построения адаптивной системы управления на основе настройки параметров указанных регуляторов. Ее внедрение не потребует значительных затрат, поскольку закон управления остается прежним. Для построения такой системы необходимо рассмотреть существующие методы настройки линейных регуляторов в электроприводах постоянного тока. Их можно разделить на классические и интеллектуальные.

Классические методы настройки параметров регуляторов требуют или наличие точной модели объекта управления [9-15], или использование тестового сигнала [16-21] для идентификации ее параметров. Получение точной, нелинейной модели объекта управления в реалиях производства является достаточно сложной задачей, а использование тестового сигнала без остановки оборудования труднодостижимо.

Однако, инженер АСУТП на производстве настраивает параметры регулятора без модели, основываясь на своем опыте, знаниях и особенностях поведения объекта. Все это может быть учтено при использовании интеллектуальных методов для решения данной задачи: метод роя частиц [22, 23], нечеткой логики [24], генетические алгоритмы [25, 26], нейронные сети [27, 28].

Каждый из этих методов имеет преимущества и недостатки. В данном случае основными требованиями к разрабатываемой системе является как необходимость ее оперативного дообучения ввиду постоянно изменяющихся параметров объекта, так и наличие знаний об особенностях ее работы (например, о наличии ограничений, накладываемых на различные сигналы). Данными характеристиками обладают

нейронные сети и экспертные системы, которые предлагается комбинировать в данном случае.

В работе [29] предложен нейросетевой настройщик параметров ПИ-регулятора. Он представляет собой базу правил, определяющую в какие моменты необходимо обучать нейронную сеть и какую скорость для отдельных нейронов использовать. Сама нейронная сеть принимает на вход данные о состоянии объекта, вычисляя на выходе необходимые в данной ситуации значения коэффициентов K_n и K_u . Данный настройщик применяется для нагревательных объектов, работающих в условиях частой смены уставок. Перспективным представляется применение подобного подхода для настройки параметров регуляторов в отдельных контурах системы управления электроприводом постоянного тока прокатной клети.

В данной работе разработана база правил для настройки параметров K_n и K_u регулятора контура тока якоря системы управления прокатной клети. Проведен численный эксперимент на упрощенной модели двухвалковой реверсивной прокатной клети. В качестве критерия эффективности работы системы использовалось суммарное энергопотребление на выполнение графика уставок.

2. Постановка задачи

В данной работе рассматривается двухвалковая реверсивная прокатная клеть с индивидуальным электроприводом каждого вала.

Крутящие моменты верхнего и нижнего привода обеспечивают два двигателя постоянного тока с независимым возбуждением. Для каждого двигателя применяется система двухзонного регулирования (рис.1).

В первой зоне (скорость вращения ниже номинальной скорости электродвигателя) применяется схема двухконтурного регулирования напряжения якоря. Во второй зоне (скорость вращения выше номинальной скорости электродвигателя) применяется схема двухконтурного регулирования тока возбуждения. Для совместной согласованной работы верхнего и нижнего привода используется схема регулирования высшего уровня, обеспечивающая заданный режим прокатки с учетом диаметров прокатных валков.

В первой зоне происходит разгон двигателей до номинальной скорости с максимальным крутящим моментом. Это обеспечивает надежный захват заготовки валками. Во второй зоне разгон двигателей от номинальной скорости происходит за счет ослабления магнитного поля статора. Такой режим работы обеспечивает достаточную производительность клети.

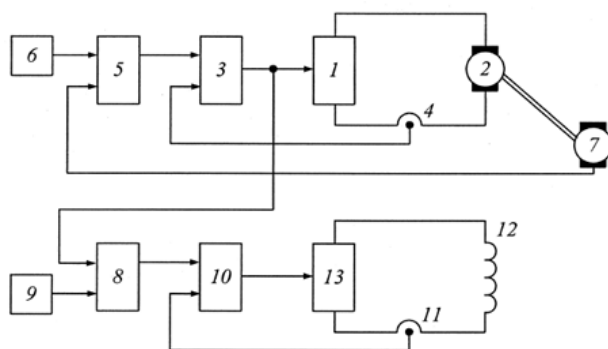


Рис. 1. – Функциональная схема электропривода двухвалковой реверсивной клети. 1– тиристорный преобразователь; 2 – якорь двигателя; 3 – регулятор тока якорной цепи; 4 – датчик тока; 5 – регулятор скорости; 6 – задатчик интенсивности; 6 – тахогенератор; 8 – регулятор ЭДС; 9 – задатчик выпрямленной ЭДС; 10 – регулятор тока возбуждения; 11 – датчик тока возбуждения; 12 – обмотка возбуждения; 13 – тиристорный возбудитель

Предметом данного исследования является контур тока в двухконтурной системе управления электроприводом постоянного тока в первой зоне. Объектом управления этого контура является обмотка якоря. Она представляется аperiodическим звеном с коэффициентом усиления K и постоянной времени T_3 . Эти параметры изменяются в ходе работы электропривода из-за нагрева обмотки и ее старения. А параметры регулятора были настроены один раз в процессе пуско-наладочных работ и в дальнейшем они не изменяются. Это приводит к тому, что с течением времени качество переходных процессов в данном контуре падает, что вызывает повышенное энергопотребление.

Нейросетевой настройщик предлагается применить для оперативной подстройки параметров ПИ-регулятора данного контура в соответствии с происходящими изменениями параметров объекта управления, но без построения его модели.

Рассмотрим структуру нейросетевого настройщика.

3. Описание нейросетевого настройщика

На рис. 2 приведена функциональная схема нейросетевого настройщика. Он состоит из:

- 1) базы правил, в условиях которых определены ситуации, когда требуется настройка параметров ПИ-регулятора. Следствиями данных правил являются значения скорости обучения отдельных нейронов выходного слоя сети;
- 2) нейронной сети, которая получает данные о задании, выходе объекта управления, формируемом ПИ-регулятором и, в данном случае, значении уставки внешнего контура по отношению к настраиваемому (уставка по скорости двигателя).

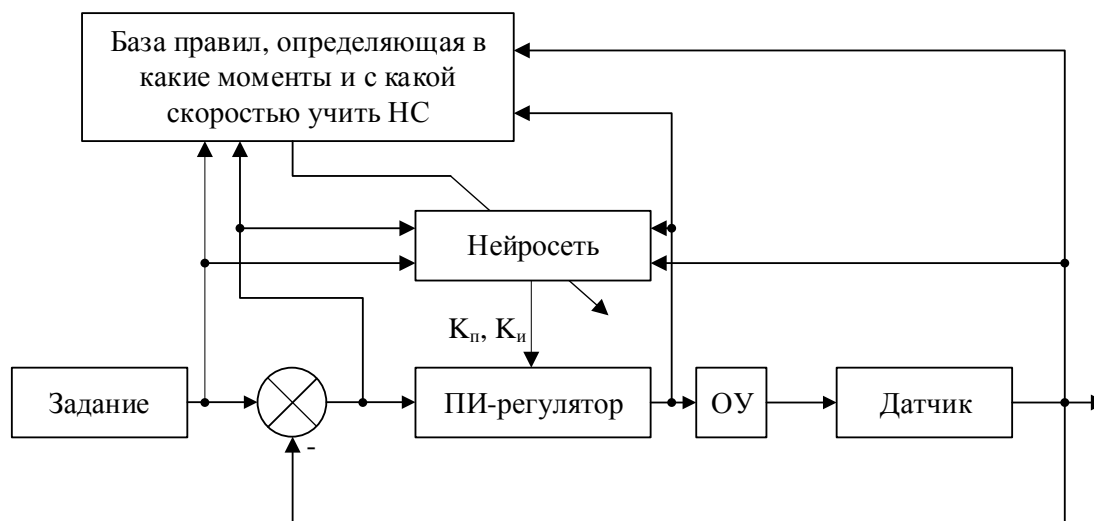


Рис. 2. – Функциональная схема нейросетевого настройщика

Рассмотрим структуру нейронной сети. Она совпадает со структурой, подобранной в работе [29] для нагревательных объектов и имеет следующий вид. Сеть состоит из трех слоев, т.к. в соответствии с [30] подобной структуры достаточно для аппроксимации функции любой сложности. Во входном слое расположено пять нейронов: уставка по току, выход контура тока, задержанный на dt секунд Δt секунд и $2\Delta t$ секунд, а также выход ПИ-регулятора тока. dt здесь – дискретность работы модели, Δt – дискретность задержки входных сигналов нейронной сети друг относительно

друга. В скрытом слое расположено 15 нейронов и сигмоидальная функция активации, в выходном слое – два нейрона, отвечающие за вычисление коэффициентов K_n и K_u , и линейная функция активации.

Значение Δt вычисляется в соответствии с методом, предложенным в работе [31]. В данном случае значение Δt составляет 1.5 мс. Обучение данной нейронной сети ведется в двух режимах:

1) обучение для задания начального состояния нейронной сети. Целью такого обучения является формирование на выходе нейронной сети коэффициентов регулятора, использовавшихся в контуре управления током до внедрения настройщика. Для проведения подобного обучения автономная обучающая выборка не требуется, данные оперативно собираются первые N секунд в оперативном режиме (N – число нейронов в скрытом слое сети).

2) непосредственно оперативное обучение нейронной сети для отслеживания изменений, происходящих в контуре тока. Производится методом обратного распространения ошибки [32]. Каждые Δt секунд происходит вызов нейросетевого настройщика и, соответственно, производится выполнение одной итерации указанного метода обучения. В соответствии с его математической моделью (1), ошибка обучения для каждого из нейронов выходного слоя имеет свое значение. Для нейрона, отвечающего за K_u , $e_2(t)$ – это текущее значение рассогласования между заданием и выходом объекта управления. Для нейрона, отвечающего за K_n , $e_1(t)$ – это значение скорости изменения данного рассогласования. При этом, прочие требования к качеству переходного процесса (кроме минимизации рассогласования) определены правилами (см. п.4).

$$\begin{aligned}\Delta w_{kj}^{(2)}(t) &= \eta_k^{(2)} \delta_k^{(2)} O_j^{(1)}, \\ \Delta b_k^{(2)}(t) &= \eta_k^{(2)} \delta_k^{(2)}, \\ \delta_k^{(2)} &= e_k \frac{dO_k^{(2)}}{ds_k^{(2)}}, k = \overline{1, 2}, \\ e_2(t) &= r(t) - y(t), \\ e_1(t) &= e_2(t) - e_2(t-1),\end{aligned}\tag{1}$$

где $r(t)$ – задание по току,

$y(t)$ – фактическое значение тока якоря,

$\eta_k^{(2)}$ – скорость обучения для k -го нейрона выходного слоя,

$\delta_k^{(2)}$ – совокупная ошибка k -го нейрона выходного слоя,

e_k – ошибка k -го нейрона выходного слоя,

$dO_j^{(2)}/ds_j^{(2)}$ – производная функции активации выходного слоя.

В данном случае используются различные скорости для всех нейронов выходного слоя, поскольку амплитудные значения K_n и K_u могут отличаться на несколько порядков. Конкретные значения скоростей обучения (η_1 и η_2) для каждого из нейронов на каждой итерации определяет база правил. Основной рекомендацией является то, что скорость обучения должна быть на несколько порядков меньше амплитудного значения соответствующего коэффициента регулятора.

Алгоритм работы настройщика представляет собой следующую последовательность действий. Если начальное состояние нейронной сети не задано, то необходимо его задать. В начале каждой итерации все скорости обучения всех нейронов приравниваются к нулю, затем происходит вызов базы правил. Если по его итогам какая-либо из скоростей обучения оказалась не нулевой, то проводится одна итерация метода обратного распространения ошибки.

4. Разработанная база правил

Скорости обучения и моменты, когда необходимо производить обучение, определяются базой правил, производящей качественные оценки реакции контура тока на задание, выработанное регулятором скорости. Задание для регулятора скорости представляет собой линейно изменяющийся сигнал, который перестает изменяться при достижении предельного значения.

4.1 Правила для определения скорости обучения нейрона выходного слоя, ответственного за вычисление K_{II}

ЕСЛИ кривая фактического значения тока пересекает кривую задания, И максимальное значение задания не было достигнуто, ТО необходимо уменьшать значение K_n , причем скорость обучения соответствующего нейрона η_1 должна быть на один порядок меньше фактического значения K_n .

ЕСЛИ кривая фактического тока якоря приближается к кривой задания по току, И максимальное значение задания не достигнуто, ТО необходимо уменьшать значение K_n , причем скорость обучения соответствующего нейрона η_1 должна быть на два порядка меньше фактического значения K_n .

ЕСЛИ модуль значения второго экстремума кривой фактического значения тока якоря больше модуля значения второго экстремума задания по току, И настройка K_u на данном переходном процессе не производилась, И кривая фактического тока и задания пересекались по крайней мере один раз, ТО необходимо увеличивать K_n , причем скорость обучения соответствующего нейрона η_1 должна быть на два порядка меньше фактического значения K_n .

ЕСЛИ модуль значения первого экстремума задания по току больше модуля значения первого экстремума кривой фактического тока в два раза, ТО необходимо увеличить значение K_n , причем скорость обучения соответствующего нейрона η_1 должна быть на один порядок меньше фактического значения K_n .

ЕСЛИ между первым и вторым экстремумом задания по току происходит пересечение данной кривой с кривой тока, ТО необходимо уменьшить значение K_n , причем скорость обучения соответствующего нейрона η_1 должна быть на два порядка меньше фактического значения K_n .

4.2 Правила для определения скорости обучения нейрона выходного слоя, ответственного за вычисление K_{II}

ЕСЛИ модуль значения первого экстремума тока больше, чем модуль значения первого экстремума кривой задания, И между первым и вторым экстремумами задания кривая тока не пересекала кривую задания, И в ходе текущего переходного процесса значение K_n не изменялось, ТО необходимо уменьшить K_u , причем скорость обучения соответствующего нейрона η_2 должна быть на три порядка меньше фактического значения K_u .

Если по результатам переходного процесса существует статическая ошибка с амплитудой более 10% текущего значения задания по току, ТО необходимо увеличить K_u , причем скорость обучения соответствующего нейрона η_2 должна быть на два порядка меньше фактического значения K_u .

Правила проверяются на срабатывание в порядке, приведенном в данной статье. В случае срабатывания какого-либо правила, оно определяет коэффициент регулятора, требующий настройки, и направление его изменения для этого переходного процесса. Это исключает вероятность срабатывания противоречащих друг другу правил.

5. Реализация нейросетевого настройщика для контура тока

Эксперимент производился на упрощенной модели двухвалковой реверсивной прокатной клетки сортопрокатного цеха №1 Оскольского Электromеталлургического Комбината. Данная прокатная клетка оснащена двигателями постоянного тока независимого возбуждения 1JW5539-5DK07-Z-001 (номинальная мощность электродвигателя 3500 кВт; пределы регулирования скорости вращения $0 \div 60/150$ об/мин; номинальный ток якоря 3865А; номинальное напряжение якоря 997 В). Целью работы является применение нейросетевого настройщика в контуре тока якоря, поэтому контур тока возбуждения не рассматривался. Построенная модель приведена на рис. 3.

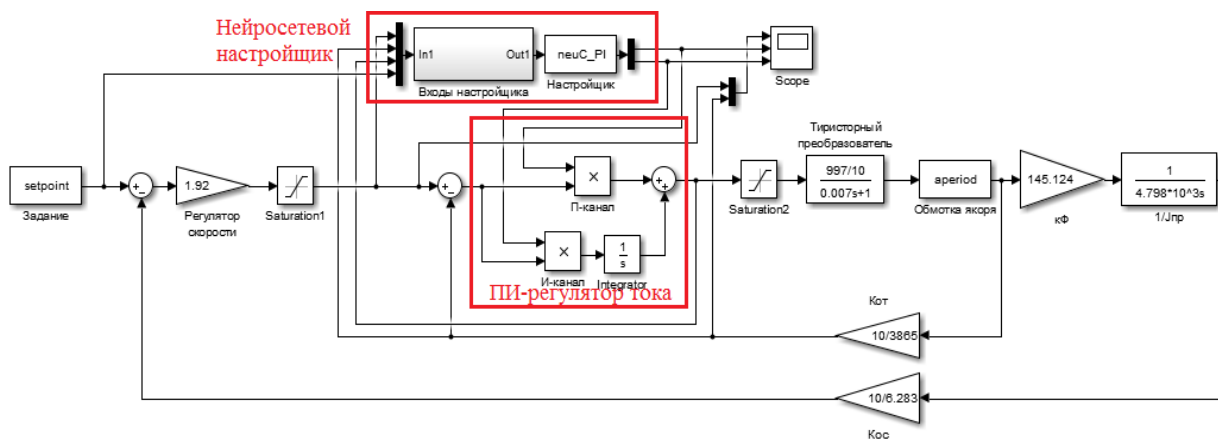


Рис. 3. – Упрощенная модель двухвалковой реверсивной прокатной клетки

Параметры регулятора скорости и регулятора тока настроены на технический оптимум. Для контура тока: $K_n = 0.24$, $K_u = 6.705$. Для контура скорости: $K_n = 1.92$. Задание по скорости реализовано в блоке *setpoint* и изменяется линейно 0 об/мин (0 В) → 60 об/мин (10 В) → 0 об/мин (0 В) → -60 об/мин (-10 В). Сигнал задания изменялся со скоростью 0.5 В/мс. С помощью блока *aperiod* реализована модель обмотки якоря в виде аperiodического звена с возможностью изменять его параметры K (номинальное значение – $1/0.024$) и T_s (номинальное значение – 0.036 с). Блоками *Saturation1* и *Saturation2* реализовано ограничение выходов регуляторов скорости и тока соответственно в пределах от -10В до 10В.

Нейросетевой настройщик также реализован в виде S-функции *NeuC_PI*. Его входами (скрыты в блоке *Subsystem* «Входы настройщика») являются текущее задание по току, выработанное регулятором скорости, фактический ток якоря (и его задержанные значения), выход регулятора тока и значения задания по скорости для определения момента реверса привода. Выходами нейросетевого настройщика являются значения K_n и K_u регулятора тока.

6. Результаты экспериментов

Было проведено две серии экспериментов. В первой до начала опыта коэффициенты ПИ-регулятора контура были изменены на следующие значения $K_n = 2$, $K_u = 2$. Задачей настройщика являлось, соблюдая все требования к переходному процессу, определенные правилами, перенастроить регулятор контура скорости. Результаты эксперимента с использованием настройщика приведены на рис. 4. В

случае, если настройщик не используется, все переходные процессы будут иметь вид, совпадающий с первым переходным процессом (рис.5).

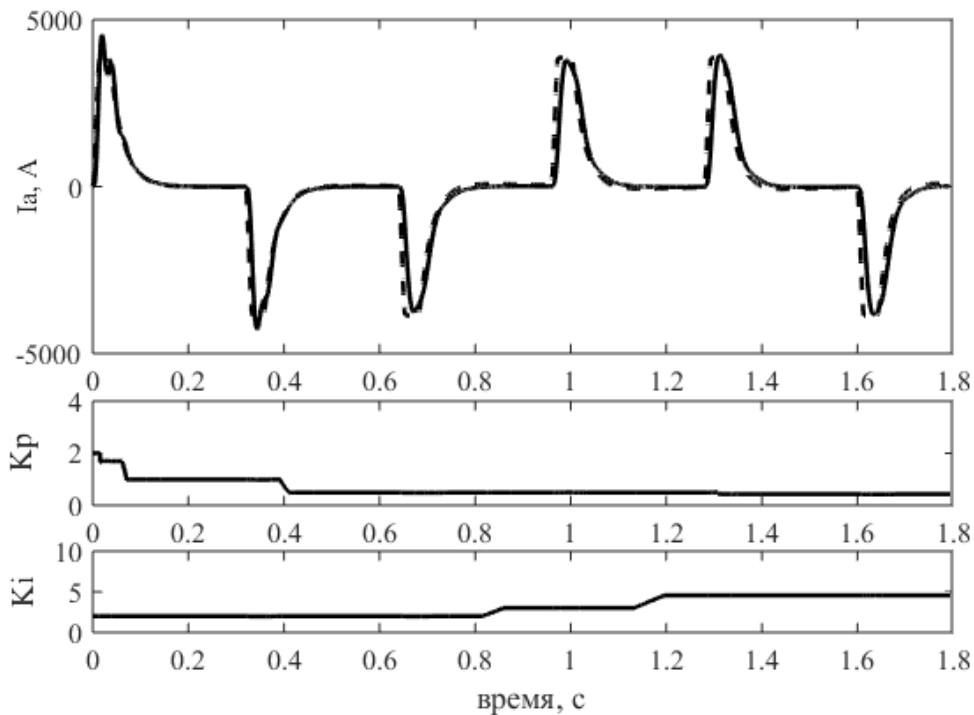


Рис. 4. – Результаты эксперимента с нейросетевым настройщиком при исходных значениях регулятора тока $K_n = 2$, $K_u = 2$

Первоначально перерегулирование составляет 15%, к шестому переходному процессу настройщику удалось свести перерегулирование к 0 процентов и убрать имевшуюся при переходном процессе колебательность. Экономия электроэнергии (энергопотребление оценивалось как суммарная мощность электропривода за время эксперимента) по итогам опыта составила 3%.

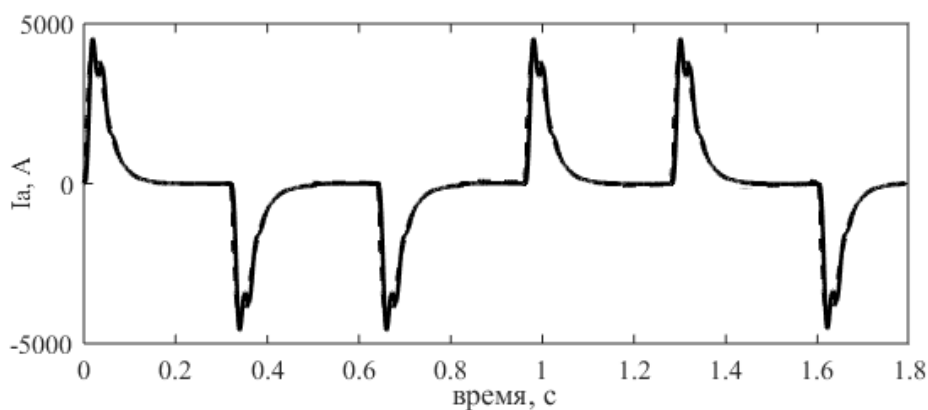


Рис. 5. – Результаты эксперимента без нейросетевого настройщика при исходных значениях регулятора тока $K_n = 2$, $K_u = 2$

Во второй серии экспериментов регулятор скорости в начальный момент имел номинальные значения своих параметров, указанные в п.5. Параметры K и T_s были увеличены в полтора раза и задачей настройщика было подобрать новые значения

коэффициентов K_n и K_u для измененного объекта управления. Результаты эксперимента с нейросетевым настройщиком приведены на рис. 6. В случае, если настройщик не используется все переходные процессы имеют вид, совпадающий с видом первого переходного процесса на рис.6 (рис.7). Для него перерегулирование составило 15,7%, переходная характеристика совершила два колебания до установившегося режима. Нейросетевому настройщику к шестому переходному процессу удалось свести перерегулирование к 0%, а переходная характеристика совершила одно колебание. Экономия электроэнергии по итогам опыта составила 4%.

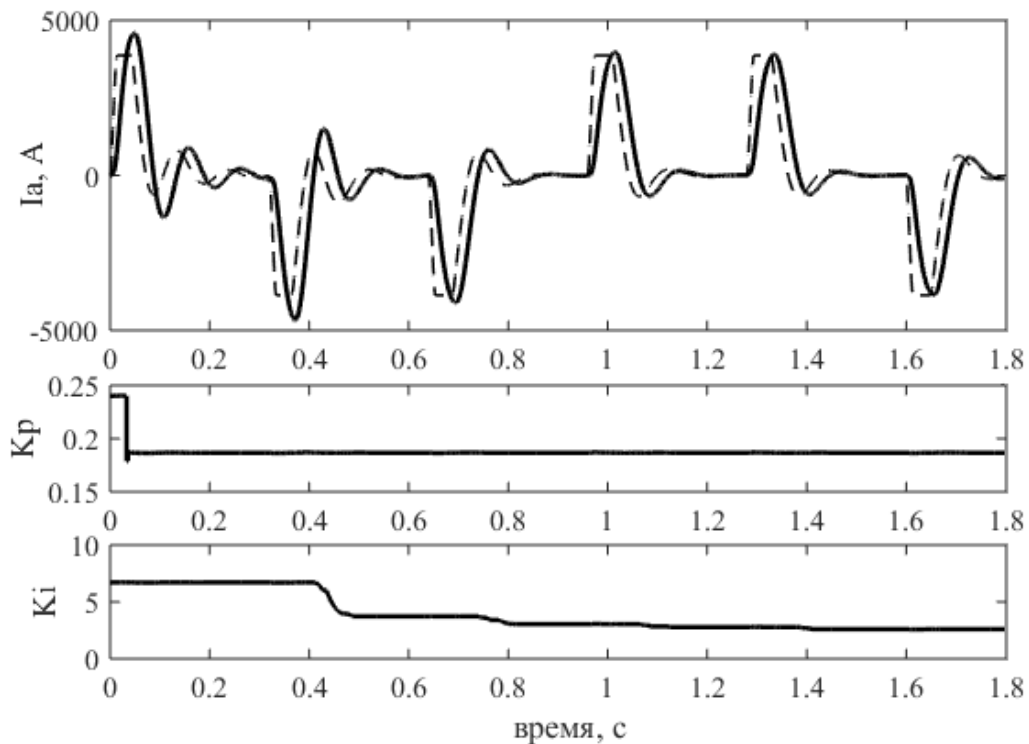


Рис. 6. – Результаты эксперимента с нейросетевым настройщиком при удвоенных значениях параметров объекта управления в контуре тока

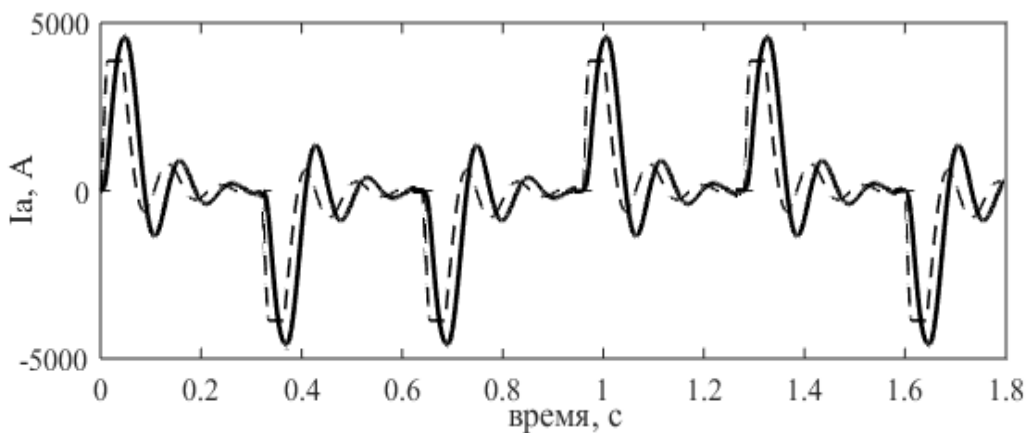


Рис. 7. – Результаты эксперимента без нейросетевого настройщика при удвоенных значениях параметрах объекта управления контура тока

7. Заключение

В результате проведенных исследований был разработан нейросетевой настройщик параметров ПИД-регулятора контура тока системы управления двухвалковой реверсивной клетью. Его применение возможно, как для настройки параметров данного регулятора в конкретный момент времени, так и для постоянного отслеживания изменений, происходящих в объекте управления в данном контуре. Результаты модельных экспериментов позволяют предположить, что его применение позволяет сократить энергопотребление прокатной клетки на 3-4%. Дальнейшие исследования связаны с совершенствованием разработанной базы правил и адаптацией настройщика для использования в контурах скорости и тока возбуждения.

Литература

1. Егорова А.О., Кузнецов В.П. Металлургический комплекс РФ: состояние, тенденции, перспективы // Фундаментальные и прикладные исследования кооперативного сектора экономики. – №2. – 2015. – С. 56-61.
2. Никифоров Б.И., Заславец Г.В. Энергосбережение на металлургических предприятиях. Монография. – Магнитогорск: МГТУ, 2000. – 283 с.
3. Храмшин В. Р. и др. Совершенствование систем управления электро-и гидроприводами широкополосного стана горячей прокатки // Электротехника. – 2015. – Т. 2. – №. 1. – С. 51-61.
4. Храмшин В. Р. Разработка и внедрение автоматизированных электроприводов и систем регулирования технологических параметров широкополосного стана горячей прокатки // Вестник ИГЭУ. – 2012. – №. 6. – С. 100-104.
5. Александров А. Г., Паленов М. В. Состояние и перспективы развития адаптивных ПИД-регуляторов в технических системах // Автоматика и телемеханика. – 2014. – №. 2. – С. 16-30.
6. Карандаев А.С., Храмшин В.Р., Лукин А.Л., Шурыгина Г.В., Головин В.В. Экспериментальные исследования тиристорных электроприводов с двухзонным регулированием скорости с улучшенными энергетическими характеристиками // Вестник ЮУрГУ. Серия: Энергетика. – 2010. – №14. – С. 67-72
7. Веренев В.В. и др. Зависимость ударных нагрузок от технического состояния клеток непрерывного прокатного стана // Металл и литье Украины. – 2009. – №1. – С. 51-54.
8. Горюнов А. Г., Ливенцов С. Н. Цифровой регулятор для системы управления с исполнительным механизмом постоянной скорости // Известия Томского политехнического университета. – 2004. – Т. 307. – №. 6. – С. 131-134.
9. Ziegler J.G., Nichols N.B. Optimum settings for automatic controllers // Transactions of the ASME. – 1942. – Vol.64. – pp. 759-768.
10. Chien K.L., Hrones J.A., Reswick J.B. On the Automatic Control of Generalized Passive Systems // Transactions of the American Society of Mechanical Engineers. – 1952. – Vol.74. – P.175-185.
11. Буков В.Н. Синтез управляющих сигналов с помощью прогнозирующей модели в адаптивной системе управления // Пробл. управления и теории информ. – 1980. – Т.9(5). – С. 329-337.
12. Петров Б.Н. Рутковский В.Ю., Крутова И.Н. Принципы построения и проектирования самонастраивающихся систем управления. – М.: Машиностроение, 1972.– 358 с.
13. Растринин Л.А. Современные принципы управления сложными объектами. – М.: Советское радио, 1980. – 232с.

14. Шендрик В.С. Синтез оптимальных управлений методом прогнозирующей модели // ДАН СССР. – 1975. – Т.224. – №3. – С. 561-562.
15. Landan J.D. Adaptive Control – The Model Reference Adaptive Control. New York: Dekker, 1980, 386 p.
16. Куропаткин П.В. Оптимальные и адаптивные системы. – М.: Высшая школа, 1980. – 287 с.
17. Растрингин Л.А. Адаптация сложных систем. – Рига: Зинатие, 1981. – 375с.
18. Растрингин Л.А. Системы экстремального управления. – М.: Наука, 1974.– 632с.
19. Растрингин Л.А. Современные принципы управления сложными объектами. – М.: Советское радио, 1980. – 232с.
20. Цыпкин Я.З. Адаптация и обучение в автоматических системах. –М.: Наука, 1968. – 326с.
21. Hjalmarsson H & Cameron M. T. Iterative feedback tuning of controllers in cold rolling mills. 14th world congress of IFAC. – 1999. – pp. 445–450.
22. Ang K., Chong G., Li Y. PID control system analysis, design, and technology. IEEE Trans. Control System Technology. – 2005. – Vol. 13, p. 559 – 576.
23. Allaoua B., Gasbaoui B., Abderrahamani A. Neuro-Fuzzy DC Motor speed Control Using Particle Swarm Optimization // Leonardo Electronic Journal of Practices and Technologies. – pp. 1583-1078.
24. Maher M.F. Algreer Design fuzzy self tuning of PID controller for chopper-fed dc motor drive // Al-Rafidain engineering. – 2008. – Vol.16. – pp.54-66.
25. Sundareswaran K., Vasu M. Genetic tuning of PI controller for speed control of DC motor drive. Proceedings of IEEE International Conference on Industrial Technology. – IEEE, 2000. – Vol. 1. – pp. 521-525.
26. Xia C. et al. Speed control of brushless DC motor using genetic algorithm based fuzzy controller. Proceeding of the 2004 International Conference on Intelligent Mechatronics and Automation, Chengdu, China, 3rd edn. A Treatise on Electricity and Magnetism. – 2004. – Vol. 2. – pp. 68-73.
27. Zhang S., Zhou X., Yang L. Adaptive PID regulator based on neural network for DC motor speed control. Conf. Electrical and Control Engineering (ICECE). – IEEE,. 2011. – pp 1950-1953.
28. Zhang, Jing, et al. PID neural network control of hydraulic roll gap control system. International Conference on Measurement, Information and Control (MIC). – IEEE, 2012. – Vol. 2.
29. Еременко Ю.И., Полещенко Д.А., Глущенко А.И. О применении нейросетевого оптимизатора параметров ПИ-регулятора для управления нагревательными печами в различных режимах работы // Управление большими системами. – 2015. – Вып.56. – С.143-175.
30. Hornik K., Stinchcombe M., White H. Multilayer feed-forward networks are universal approximators // Neural networks. – 1989. – Vol. 2. – No. 3. – P. 359-366.
31. Еременко Ю.И., Полещенко Д.А., Глущенко А.И. О разработке метода определения значений задержек входных сигналов нейронной сети при реализации нейросетевого оптимизатора параметров ПИ-регулятора // Известия ЮФУ. Технические науки. – 2015. – №10. – С. 157-168.
32. Omatu S., Khalid M., Yusof R. Neuro–Control and its Applications. – London:Springer, 1995. – 255 p.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант 15-07-0609215) и фонда Президента РФ (грант №14.Y30.15.4865-МК).

АЛГОРИТМ ИНТЕГРАЦИИ ЗАДАЧ С ГРАФИЧЕСКИМ ТИПОМ ВВОДА И ОЦЕНКА ИХ РЕШЕНИЙ В СИСТЕМАХ УПРАВЛЕНИЯ ОБУЧЕНИЕМ

Спиридонов Р.С.

Московский Физико-технический Институт,
141700 г. Долгопрудный, Московская область, romars@phystech.edu

Аннотация

Решение практических задач является ключевым звеном образовательного процесса. Однако в современных системах управления обучением отсутствует единый подход к моделированию задач для самопроверки и тестов. Предложен метод моделирования объектов типа «практическая задача», который позволяет создавать алгоритмы оценки ответов студентов различного типа, что экономит время преподавателя. Разработан алгоритм оценки ответов для графических типов ввода. Выполнена реализация метода в виде независимого веб-сервиса, позволяющего интегрировать данную функциональность в системы управления обучением.

1. Введение

Важным звеном процесса обучения является решение практических задач. Именно на основе этого звена путем интенсивной практики *декларативные* знания, т.е. знания фактов, явлений и закономерностей, переходят в *процедуральные* знания, т.е. способность решать реальные задачи. Наличие процедуральных знаний отличает профессионалов предметной области от новичков [1].

В то время как для получения декларативных знаний достаточно качественных текстовых и мультимедиа материалов (электронные книги, видео-лекции и т.п.), формирование навыков решения задач требует более сложных программных средств и методик [2]. Одно из ключевых отличий заключается в необходимости предоставлять обратную связь студенту о правильности решения задачи, причем процесс проверки правильности может отнимать у преподавателя достаточно большое время. Таким образом, возможность автоматизировать оценку правильности ответов могла бы сэкономить время преподавателя и позволить перераспределить его в пользу, например, расширенных консультаций студентов или корректировку плана их обучения на основе уже имеющихся результатов оценки.

Стоит отметить, что педагогический интерес представляют именно сложные типы вводы. Как показано в работе [3], решение задач, в которых предлагается выбор одного из заранее выданных ответов, не способствуют развитию у обучающихся творческих подходов к решению сложных задач.

На сегодняшний день активно в учебных заведениях активно используются различные системы управления обучением (в дальнейшем, СУО). Наиболее популярной в российской среде является система Moodle[4], но можно так же выделить системы Sakai и Ilias[5]. Несмотря на то что эти системы предлагают расширенный набор возможностей по работе с группой студентов, их функциональность по поддержке реализации в цифровом виде практических задач со сложными типами ввода очень ограничена[6,7]. Кроме того, различные системы представляют различные форматы данных для работы с практическими задачами, что не позволяет, например, легко использовать задачи из одной системы в другой.

Таким образом, становится актуальной проблема создания набора функций по автоматизации проверки правильности решения практических задач [8–10], а также по интеграции перечисленной функциональности в произвольные СУО [11,12].

2. Описание метода

Рассмотрим задачу реализации независимого приложения, которое производило бы отображение практической задачи, обеспечивал взаимодействие со студентом посредством ввода им ответа, и автоматизированную проверку его правильности.

Формализуем втерминахязыкаXMLобъект «практическая задача», с которым будет работать приложение. В соответствии с выбранной моделью на рис. 1, описывается следующим набором свойств:

1. `<valuename='specification'>` - условие задачи, заданное в виде HTMLразметки вопроса со вставками зарезервированных тегов для обозначения специальных элементов, обрабатываемых . Примерами таких зарезервированных тегов являются `<fieldtype="{тип_вопроса}">` (поле для ответа) и `<latex>` (формула на языке LaTeX).
2. `<valuename='correctN'>` - правильные ответы к полю `<field>` порядкового номера N, где каждый ответ задается в содержимом тега `<correct method="{алгоритм_оценки}">...</correct>`; здесь автор задачи описывает правильный ответ и алгоритм оценки.

Таким образом, задача автоматизации оценки правильности ответа сводится к созданию алгоритма сравнения ответа, введенного студентом с правильным ответом, заданным автором задачи (при этом в задаче может быть один или несколько вариантов правильных ответов).

```

<item>
<value name='specification'>
<!-- CUSTOM HTML OF QUESTION WITH ANSWER FIELDS -->
Draw the graph of <latex>y=x^2</latex>.
<field type='graph' maxx='6' minx='-6' maxy='4' miny='-4'><!-- FIELD 1 -->
</value>
<value name='correct1'>
<!-- CORRECT ANSWER FOR FIELD 1 -->
<correct method="analytical" score="1" tolerance="0.01">
[{"type":"parabola-x","dashed":true,"points":[
  {"coord":[0,0],"label":"","empty":false},
  {"coord":[2,4],"label":"","empty":false,"dotted":false}]]
</correct>
</value>
</item>

```

На базе предложенной модели было разработано приложение на языке Java, работающее как веб-сервис. Реализация описанного подхода в виде веб-сервиса означает возможность развертывания приложения на отдельном сервере или кластере, что дает ряд преимуществ:

- Одна версия приложения может быть использована сразу несколькими потребителями;
- В случае большого числа потребителей и запросов, есть возможность производить горизонтальное масштабирование (увеличение число серверов приложения) и балансировку нагрузки (равномерное распределение нагрузки между серверами приложения) для повышения отказоустойчивости;
- Обеспечивается единый жизненный цикл и версионирование приложения, что упрощает его поддержку и процедуру обновления.

Для обеспечения возможности интеграции с СУО разработан программный интерфейс API, основанный на протоколе REST[13]. API вызов по протоколу REST позволяет максимально упростить работу с веб-приложением, поскольку вызов API функции происходит так же, как обращение к странице веб-сайта по URL с передачей данных по протоколу HTTP. В случае необходимости получить данные применяется метод GET, в случае отправки большого объема данных – метод POST. В параметрах HTTP запроса указываются параметры API функций. Обработчики запросов реализованы как сервлеты на языке JSP, а само приложение развернуто в контейнере сервлетов Tomcat[14].

В соответствии с принципом инкапсуляции, взаимодействие с веб-сервисом осуществляется только посредством вызова функций API, обеспечивающих следующие возможности:

1. **String[] create(String[] itemXML, [options])** – инициализация (подготовка) задачи; функция возвращает уникальный идентификатор подготовленной к отображению задачи (ID).
2. **JSONrender(String[] id, [options])** – отображение задачи с идентификатором ID; функция возвращает JSON код ранее подготовленной задачи, который может быть однозначным образом транслирован в HTML код в веб-браузере и отображен студенту.
3. **JSONanswer(String answer)** – оценка правильности ответа студента, передаваемого в переменной answer; функция возвращает результат оценки в виде числа от 0 до 1 (0 – неверно, 1 – верно, промежуточные значения – частично верно), а так же дополнительную обратную связь (например, подсказку или обоснование выставленной оценки). Стоит учесть, что в одной задаче может быть несколько что в одной задаче может быть несколько полей для ввода, поэтому и ответов может быть несколько. Поэтому возвращаемый результат закодирован в формате JSON следующего вида:

```
{ score: 0.5, rejoinder: [ "Your answer was partially correct." ],
  subGrades: { "1": { score: 1.0, rejoinder: [ "Correct." ] }
               "2": { score: 0.0, rejoinder: [ "Incorrect." ] } } }
```

Предложенный подход предполагает наличие трех видов пользователей:

1. *Авторзадачи*: создает XML код задачи и правильные ответы к ней, прикрепляет необходимые мультимедиа ресурсы (картинки, формулы и т.п.);
2. *СУО* : вызывает API-функции разработанного приложения, т.е. создания (create), отображения (render), и получения результатов оценки (answer);
3. *Студент*: взаимодействует с СУО через веб-браузер; входит на страницу домашнего задания или экзамена, затем, после отображения задачи, вводит и отправляет на сервер свой ответ.

Рассмотренную модель можно расширить для поддержки нескольких вариантов одной задачи путем добавления блока с описанием алгоритма ее решения по произвольным входным данным [15].

3. Автоматизации оценки графических типов ввода

Рассмотрим приложение данного метода к решению конкретной задачи по автоматизации оценки ответов типа «график» (`<fieldtype="graph">`). Основной задачей является реализация алгоритма сравнения правильного графика, заданного автором задачи, и введенного студентом графика.

Для начала, определим формат сериализации объектов типа «график». Безусловно, на сегодняшний день существует множество решений, в том числе с открытым исходным кодом, поддерживающих рисование и сериализацию графиков в веб-интерфейсе. В данной работе используется `jsxgraph[]`, формат графиков которых описывается в формате JSON (см. Рис. 1, содержимое тега `<correct>`), содержащем указание типа графика (`type="parabola-x"`) и нескольких его точек (`points:[{coord:[0,0.5],coord:[1,1.5]}`]).

Таким образом, задача оценки правильности ответа сводится к сравнению двух объектов: JSON представления правильного ответа и ответа студента. Опишем реализованный алгоритм оценки графических ответов на примере сравнения двух графиков парабол, где правильным ответом является парабола $y = x^2$.

Входные параметры алгоритма (подставлены цифры из примера на Рис. 1):

- $[(\min x, \min y), (\max x, \max y)] = [(-6, -4), (6, 4)]$ – диапазон для сравнения графиков (он же – видимая область для ввода графика у студента)
- `tolerance = 0.01` (или 1%) – допустимая погрешность ответа в % от длины видимого диапазона (по x по y может быть различная абсолютная погрешность, если диапазон прямоугольный).

Возвращаемое значение: `score` $\in [0, 1]$ – оценка правильности.

Описание алгоритма:

1. График студента и правильный график разбивается на множество точек с шагом дискретизации $\delta_x = (\max x - \min x) \times \text{tolerance}$ по x , $\delta_y = (\max y - \min y) \times \text{tolerance}$ по y . Шаг дискретизации по x и по y может быть различным, если масштаб по осям x и y различный – в этом случае вся

видимая область дополняется прямоугольной сеткой, ячейками которой будут прямоугольники с длиной стороны, равной указанной автором точности в виде % от видимого диапазона.

- Для каждой точки из графика студента производится поиск ближайшей точки из графика правильного ответа – последняя должна попасть в эллиптическую окрестность вида

$$\frac{(x - x_c)^2}{\delta_x^2} + \frac{(y - y_c)^2}{\delta_y^2} \leq 1 \quad (1)$$

В формуле выше: (x_c, y_c) – координаты точки правильного ответа, а (x, y) – координаты рассматриваемой точки в ответе студента. Если точка правильного ответа находится в указанном эллипсе, то точка ответа студента считается правильной. Иначе, неправильной.

- Возвращается оценка ответа, равная доле правильных точек ответа студента в общем числе точек N :

$$score = \frac{\sum_{i=1}^N \text{ifthen} \left(\frac{(x_i - x_c)^2}{\delta_x^2} + \frac{(y_i - y_c)^2}{\delta_y^2} \leq 1; 1; 0 \right)}{N} \quad (2)$$

Использование дискретизации обусловлено двумя факторами: с одной стороны, дискретизация абсолютно необходима для эффективной сложности алгоритма $O(N^2)$; с другой стороны, мы не можем требовать от студента абсолютной точности в вводе ответа в силу естественной погрешности при использовании устройства для ввода.

Традиционно приемлемой считается погрешность в половину цены деления. Допустим, что задан график в координатной сетке $[(0,0); (10,10)]$ и цена деления равна 0.2. Тогда допустимая погрешность будет 0.1, что соответствует 1% от видимого диапазона, т.е. входному параметру $tolerance = 1\%$. При таком значении входного параметра, число точек дискретизации на одном графике $N \sim 100$, что соответствует $100^2 = 10,000$ операций сравнения, что обеспечивает эффективное выполнение алгоритма даже при большой загрузке на систему. Современные процессоры на персональных компьютерах позволяют вычислять порядку 10^9 операций с плавающей точкой в секунду.

4. Заключение

Разработан метод моделирования практических задач и алгоритмов их оценки, позволяющем интегрировать задачи в различных системах управления обучением. Реализовано приложение в виде веб-сервиса, обеспечивающего интеграцию функциональности по работе с задачами в произвольные системы посредством RESTAPI протокола. Данный протокол предоставляет возможность инициализировать задачи, отображать их и проверять правильность ответа студента на эту задачу. Данный подход позволяет сэкономить ценное время работы преподавателя благодаря автоматизации процесса выставления оценки решенным задачам в домашних работах и тестах. На базе данного метода разработан алгоритм оценки ответов на вопросы, требующие ввод графика.

Литература

1. О. И. Ларичев, Е. В. Нарыжный, В. П. Кузнецова, Э. И. Брук. Новые возможности компьютерного обучения // Вестник Российской Академии Наук. 1999. Т. 69, № 2. С. 106–111.
2. Ларичев О. И. Теория подсознательных решающих правил–новый взгляд на экспертное мышление // Труды восьмой национальной конференции по искусственному интеллекту с международным участием. 2002. Т. 1. С. 1–14
3. Becker J.P., Shimada S. The open-ended approach: A new proposal for teaching mathematics. Reston, 2005.
4. Белозубова А., Николаев Д. Система дистанционного обучения Moodle. Учебно-методическое пособие // СПб. СПбГУИТМО. 2007
5. Богомолов В.А. Обзор систем управления обучением // Образовательные технологии и общество. 2007. Т. 10, № 3.
6. Question types // Moodle 2.9 documentation [Электронный ресурс]. URL: https://docs.moodle.org/29/en/Question_types (дата обращения: 23.05.2015).
7. SAMigo Features // Sakai Wiki [Электронный ресурс]. URL: <https://confluence.sakaiproject.org/display/SAM/SAMigo+Features> (дата обращения: 23.05.2015).
8. Williamson D.M., Bejar I.I., Mislevy R.J. Automated scoring of complex tasks in computer-based testing. Manwah, New Jersey: Laurence Erlbaum Associates, Inc., 2006.
9. Livne N., Livne O., Wight C. Can automated scoring surpass hand grading of students' constructed responses and error patterns in mathematics // MERLOT J. Online Learn. 2007. Т. 3, № 3. С. 295–306.
10. Fife J.H. Automated scoring of mathematics tasks in the common core era: enhancements to m-rater in support of cbal TM mathematics and the common core assessments // ETS Res. Rep. Ser. 2013. Т. 2. С. 1–35.
11. Гурченков А.А., Спиридонов Р. С. Метод моделирования задач для организации автоматизированной проверки ответов студентов в системах управления обучением // Наука и образование: электронное научно-техническое издание. — 2015. — №11. — С. 426-442.
12. Спиридонов Р.С. Метод интеграции алгоритмов оценки ответов студентов в системах электронного обучения // Перспективы развития информационных технологий: сборник материалов XXV Международной научно-практической конференции. — Новосибирск: Издательство ЦРНС, 2015. — С. 114-119.
13. Richardson L., Ruby S. RESTful Web Services // Vasa. 2008. 440 с.
14. Перри Б.У. Развертывание одиночной страницы JSP под Tomcat // Java сервлеты и JSP. Сборник рецептов. 2-е изд. Москва: КУДИЦ-Образ, 2006. С. 41–43.
15. Спиридонов Р.С. Синтез уникальных вариантов домашних заданий с помощью КС-грамматик в системах электронного обучения // Инновации на основе информационных и коммуникационных технологий: Материалы международной научно-практической конференции. — М.: НИУ ВШЭ, 2015. — С. 112-114

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ СЛЕЖЕНИЯ БОРТОВОЙ ВИДЕОКАМЕРОЙ ЛЕТАТЕЛЬНОГО АППАРАТА ЗА ДИНАМИЧЕСКОЙ ЦЕЛЬЮ В ВОЗМУЩЕННОЙ СРЕДЕ

М.В. Хачумов

Институт системного анализа Федерального исследовательского центра «Информатика и управление» Российской академии наук, Москва
khmike@inbox.ru

Аннотация

Рассматривается задача управления двухкоординатной бортовой видеокамерой в процессе следования летательного аппарата (ЛА) по заданному маршруту. ЛА играет роль подвижной управляемой платформы, на которую воздействует ветровая нагрузка, что мешает точному наведению камеры на заданную цель. При моделировании системы «ЛА-видеокамера» используются алгоритмы интеллектуального управления, имитирующие действия летчика-оператора, а также структурные схемы автоматического управления ЛА и видеокамеры, построенные на основе передаточных функций. Моделирование осуществляется средствами программного обеспечения MATLAB и системы Simulink. Исследуются точностные характеристики, отражающие способность системы управления камерой реагировать на поведение динамической цели.

1. Введение

Предполагается, что на летательном аппарате (ЛА) установлена видеокамера, которая осуществляет слежение за подвижным объектом в условиях возмущений. Задачи управления движением беспилотного ЛА в процессе отработки различных полетных заданий, в том числе планирования прохождения через заданные опорные точки в требуемые моменты времени с учетом ограничений на переменные состояния и управления были рассмотрены в работах [1, 2, 3]. Методы управления видеокамерой, установленной на борту ЛА, изучались в трудах отечественных и зарубежных ученых С.П. Фирсова, А.В. Хамухина, Н.В. Птицына, S. Youn, L.S. Davis. Задача высокоточного наведения бортовой камеры и слежения за подвижными объектами в автоматическом режиме была рассмотрена в работе [4].

Целью настоящего исследования является изучение точности наведения видеокамеры в системе «ЛА-видеокамера» на основе выбранных математических моделей с учетом действующих возмущений в процессе слежения за динамическим объектом в реальном времени. Предполагается, что задача следования летательного аппарата по заданному маршруту в возмущенной среде решается в соответствии со стратегиями, имитирующими поведение летчика. Стратегии реализуются наборами правил при ограничениях на управление с учетом действующих возмущений и в настоящей работе не приводятся, так как они описаны в основополагающей работе [5].

Предлагаемый подход позволяет достичь приемлемых результатов управления системой «ЛА-видеокамера» при решении задачи наблюдения за динамическими объектами простыми средствами, что существенно для реализации режима реального времени непосредственно на борту автономного ЛА.

2. Модель видеокамеры

В качестве видеокамеры рассматривается поворотная роботизированная PTZ-камера (Pan Tilt Zoom), которая позволяет эффективно решать задачи видеонаблюдения. Быстрый переход от одной точки наблюдения к другой достигается за счет одновременного движения по азимуту (на $0 \div 360$ градусов) и по углу места (на $0 \div 90$ градусов).

Для симуляции работы видеокамеры в системе Simulink использованы типовые схемы систем управления ее горизонтальным и вертикальным приводами [6], показанные на рис. 1.

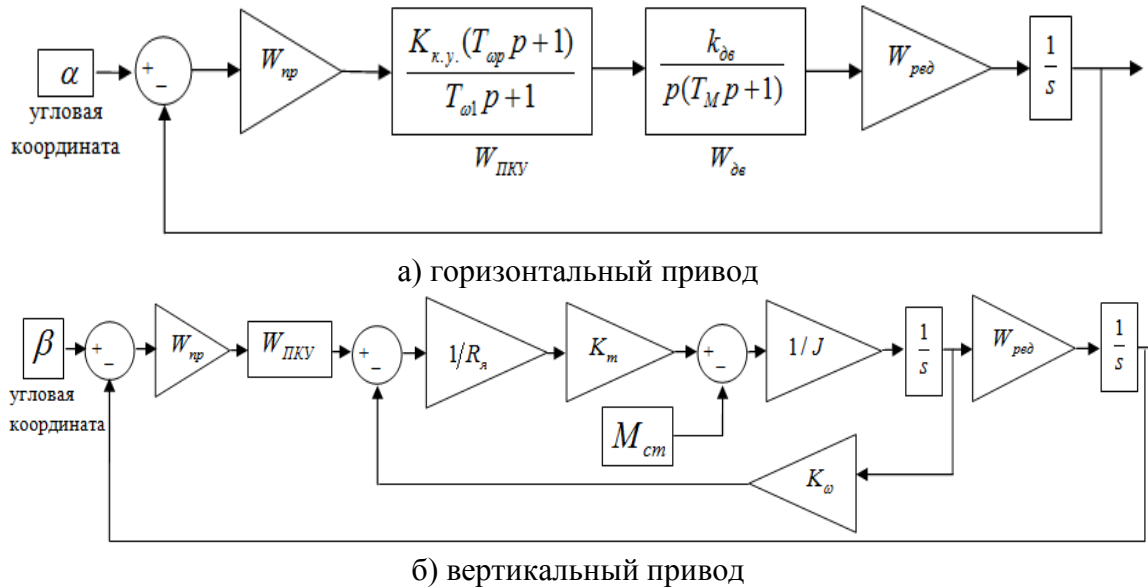


Рис. 1. – Модель приводов наведения видеокамеры

Система автоматического управления видеокамеры в принятой модели имеет следующие составляющие: $W_{\text{дв}} = \frac{k_{\text{дв}}}{p(T_M p + 1)}$ – передаточная функция двигателя; $k_{\text{дв}}$ – коэффициент передачи двигателя; T_M – электромеханическая постоянная времени двигателя; $W_{\text{ред}}$ – передаточная функция редуктора; $W_{\text{пр}}$ – передаточная функция преобразователя; $M_{\text{ст}}$ – статический момент, приведенный к валу двигателя; $W_{\text{ПКУ}} = \frac{K_{\text{к.у.}}(T_{\text{оп}} p + 1)}{T_{\text{ол}} p + 1}$ – передаточная функция последовательно корректирующего устройства (ПКУ); $K_{\text{к.у.}}$ – коэффициента усиления ПКУ; $T_{\text{ол}}, T_{\text{оп}}$ – постоянные, отражающие инерционные свойства звена ПКУ; $R_{\text{я}}$ – сопротивление якоря; K_m – электромагнитный коэффициент; J – момент инерции вращающихся частей, приведенный к валу двигателя.

Рассмотренные схемы (Рис. 1) были использованы для моделирования автоматической стабилизации приводов видеокамеры в задаче слежения за динамическим объектом.

3. Модель платформы «ЛА-видеокамера»

В работе [7] была промоделирована работа системы автоматического управления «ЛА-видеокамера», осуществляющей полет по заданному маршруту в условиях ветровой нагрузки для плоского случая. В качестве продолжения начатых исследований в настоящей работе была изучена модель динамической системы для пространственного случая. Рассмотрим назначение основных блоков модели «ЛА-видеокамера», построенных на основе передаточных функций звеньев в системе MATLAB Simulink (Рис. 2).

Блоки «Тангаж» и «Рыскание». На вход блоков подаются углы, которые должен отработать ЛА с учетом возмущений. На выходе блоки выдают текущие углы тангажа и рыскания. Структурные схемы стабилизации углов были рассмотрены в работе [5].

Блок «Вычисление координат». В принятой модели положение ЛА в текущий момент времени t вычисляется на основе координат в предыдущий момент времени $t - 1$ и известных параметров: заданной скорости ЛА, углов тангажа и рыскания, а также скорости и направления ветра.

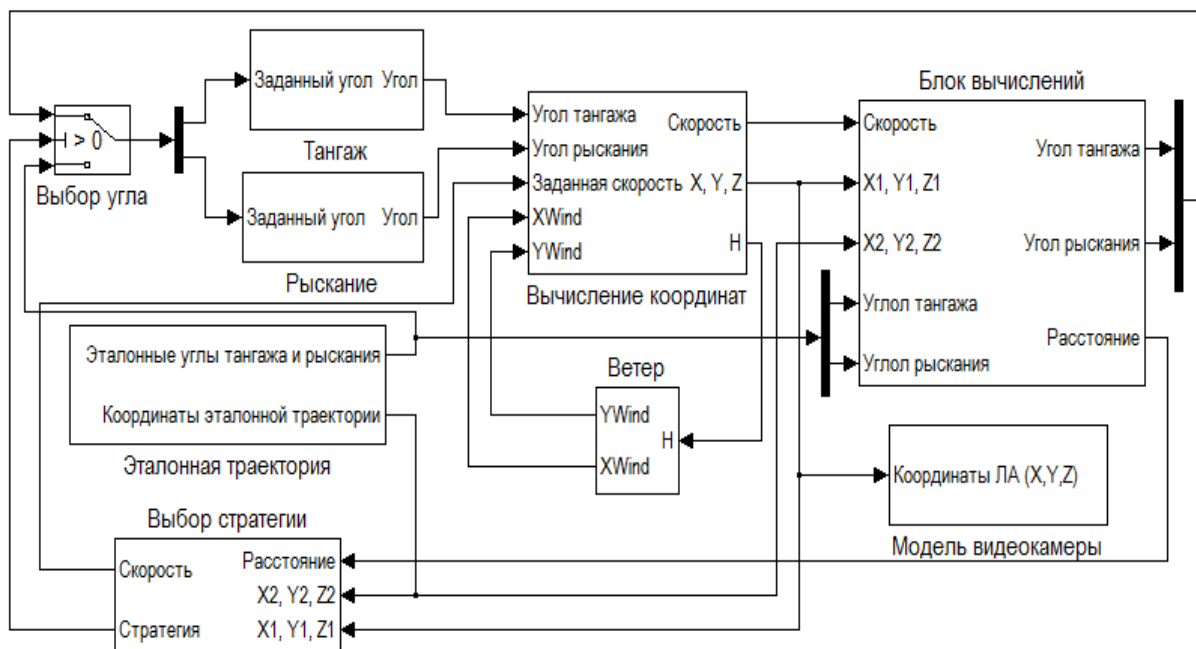


Рис. 2. – Структурная схема моделирования динамической системы «ЛА-видеокамера»

Блок «Ветер» моделирует ветровые воздействия с помощью стандартной функций MATLAB Wind Shear Model.

Блок «Эталонная траектория» выдает текущие координаты и углы ориентации эталонного движения.

«Блок вычислений» рассчитывает углы для сближения с эталонной траекторией.

Блок «Модель видеокамеры» (отдельно показан на Рис. 3) рассчитывает углы ориентации и осуществляет управление бортовой видеокамерой, в соответствии с текущим положением ЛА и объекта наблюдения.

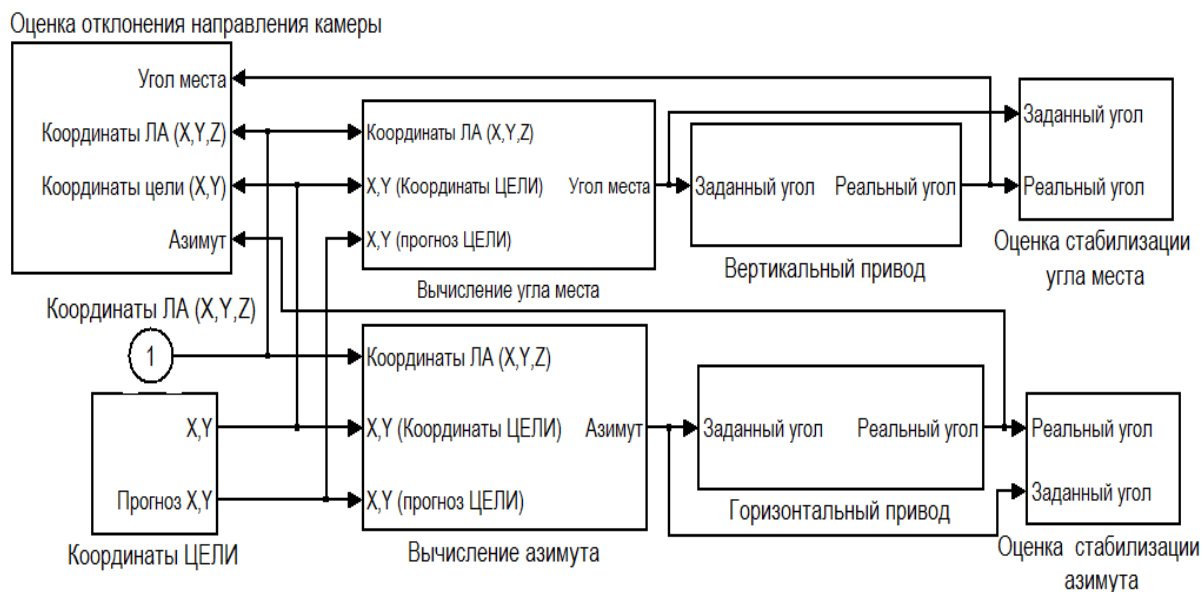


Рис. 3. – Структурная схема управления видеокамерой для наблюдения за целью

Блок «*Выбор стратегии*» вырабатывает конкретные управления, в соответствии с текущим состоянием системы.

Рассмотрим основные блоки структурной схемы управления видеокамерой.

Блоки «*Вычисление угла места*» и «*Вычисление азимута*» рассчитывают углы ориентации видеокамеры для наведения на объект. Заметим, что аналитическое решение задачи высокоточного нацеливания видеокамеры, находящейся в произвольном положении, на некоторую точку, расположенную на плоскости наблюдения содержится в работе [8].

Блоки «*Вертикальный привод*» и «*Горизонтальный привод*». На вход блоков подаются углы, которые должна отработать бортовая видеокамера. На выходе блоки выдают текущие углы ориентации видеокамеры.

Блок «*Координаты ЦЕЛИ*» выдает текущие координаты наблюдаемого объекта. Для опережающего наведения видеокамеры блок прогнозирует траекторию движения объекта. Отметим, что в работе [9] для прогнозирования значений временного ряда предложено использовать искусственную нейронную сеть прямого распространения с сигмоидальной функцией активации.

Блок «*Оценка отклонения направления камеры*» рассчитывает отклонение реально наблюдаемой точки от заданной.

Система, реализованная в SIMULINK (Рис. 2., Рис. 3), предназначена, таким образом, для моделирования полета ЛА с установленной на борту видеокамерой в условиях ветровой нагрузки. Она послужила основой для решения задач следования по заданному маршруту и наблюдения за некоторой целью в рамках единой системы.

4. Экспериментальные исследования

Проводилось моделирование процесса слежения за динамическим объектом, осуществляющим движение на плоскости наблюдения, с помощью бортовой камеры, закрепленной на летательном аппарате. ЛА следовал по заданному пространственному маршруту в неспокойной атмосфере при воздействии встречного ветра. Ниже на Рис. 4 - Рис. 6 приведены результаты и характеристики этого эксперимента.

На Рис. 4 приведен график разности между расчетными и реальными углами наблюдения за объектом для горизонтального привода. При этом максимальное расхождение между углами после стабилизации наведения видеокамеры (через 2 сек.) составило 0.03 градусов.

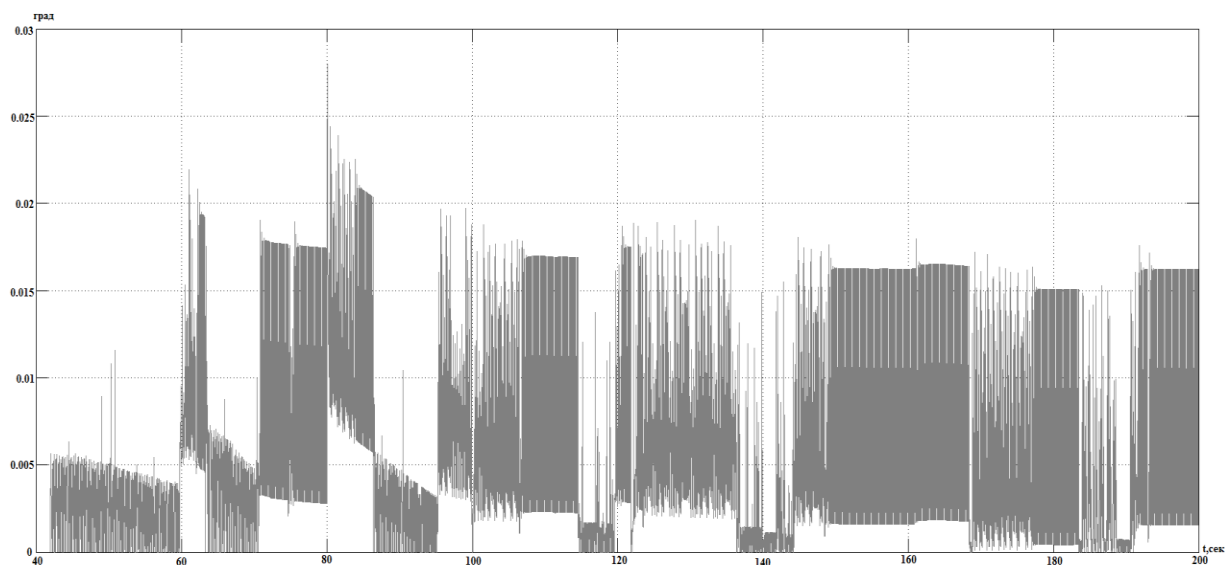


Рис. 4. – Стабилизация горизонтального привода видеокамеры

На Рис. 5 представлен график расхождения местоположения центра наблюдаемой видеокамерой области от расположения центра объекта. Максимальное расхождение составляет 4.1 м.

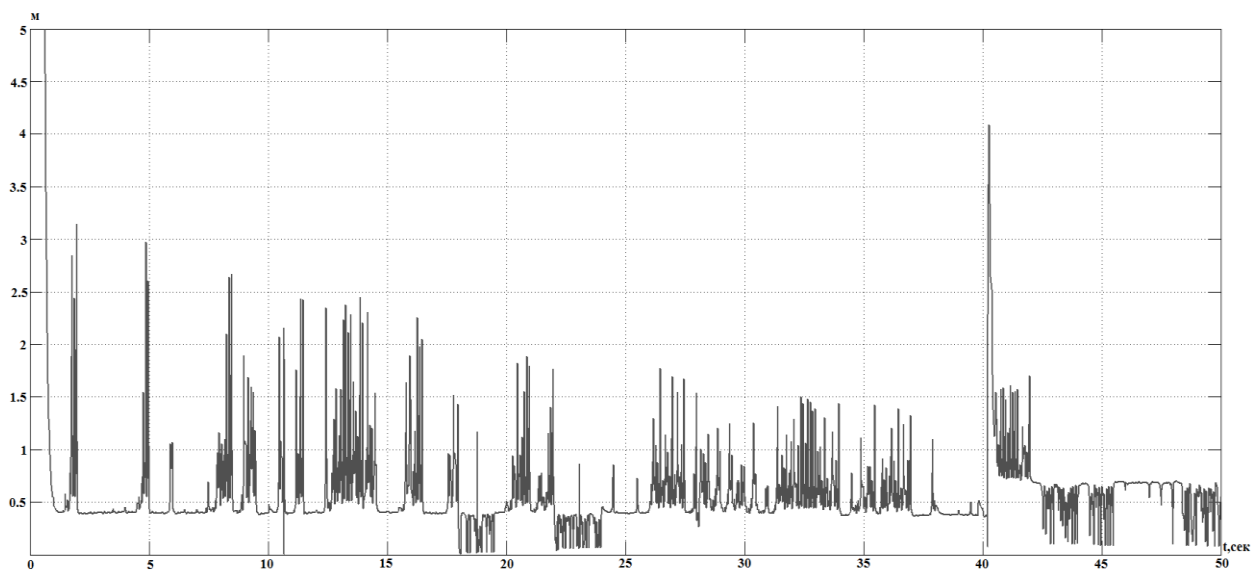


Рис. 5. – Расхождение с точкой наблюдения

На Рис. 6 показаны траектории движения ЛА и наблюдаемого динамического объекта. Траектория движения объекта на плоскости отмечена светлой кривой. Траектория движения ЛА, следующего по заданному маршруту – темной кривой. Начальные точки движения объектов показаны окружностями.

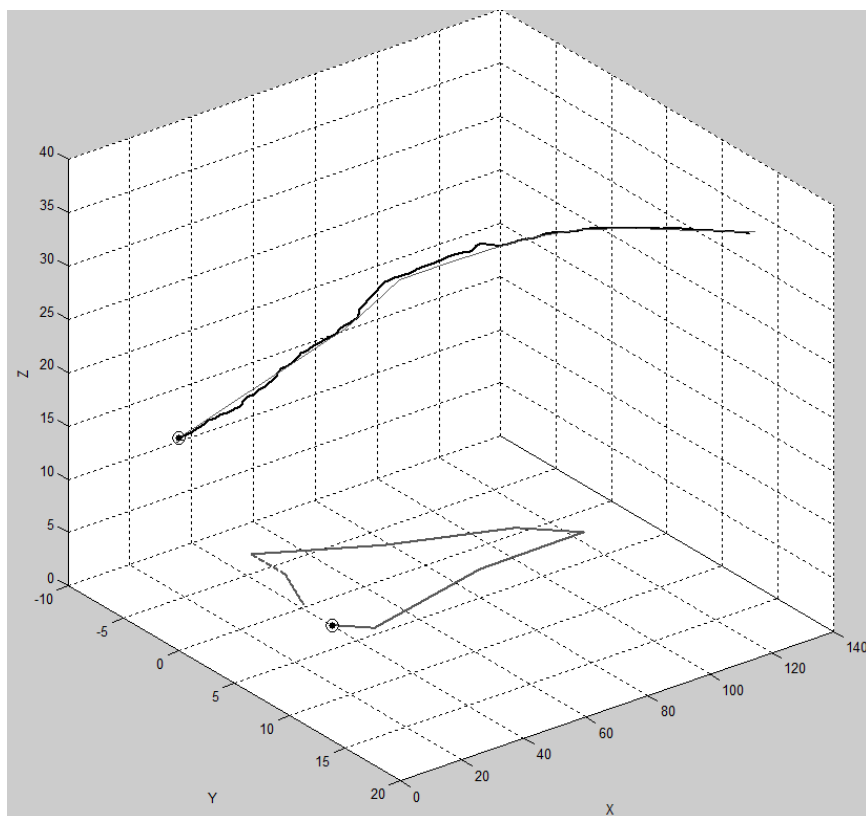


Рис. 6. – Траектории движения динамического объекта и ЛА

В результате выполненных экспериментальных исследований показано, что предложенный авторами в работе [7] алгоритм интеллектуального управления ЛА в сочетании с алгоритмом управления видеокамерой полностью обеспечивают выполнение тактико-технических требований, предъявляемых к системе «ЛА-видеокамера» по функциональным возможностям, точности наведения.

5. Заключение

Исследовано поведение динамической системы «ЛА-видеокамера» в процессе решения одновременно двух задач: следование по заданному маршруту при наличии ветровых нагрузок и слежение за динамическим объектом. Путем моделирования средствами MATLAB установлено, что видеокамера достаточно уверенно обрабатывает задачу наблюдения за динамическим объектом в возмущенной среде. При этом в выполненных экспериментах ошибка отклонения центра наблюдаемой области от центра объекта (через 2 сек. после начала наблюдения) не превосходила величины 4.1 м., а максимальное расхождение между расчетным и реальным азимутом после стабилизации привода (через 2 сек.) не превышало 0.03 градусов. *Работа выполнена в рамках проекта РФФИ № 15-07-00925.*

Литература

1. Каршаков Е.В. Особенности алгоритмов управления летательным аппаратом при аэросъемке. — Проблемы управления, № 3, 2012, с. 71–76.
2. Канатников А.Н., Крищенко А.П., Ткачев С.Б. Допустимые пространственные траектории беспилотного летательного аппарата в вертикальной плоскости. — Элек-

- тронный журнал «Наука и образование», № 3, 2012. URL: <http://technomag.bmstu.ru/doc/367724.html> (дата обращения 21.03.2016).
3. Крищенко А.П., Канатников А.Н., Ткачев С.Б. Планирование пространственного разворота беспилотного летательного аппарата. — Вестник МГТУ им. Н.Э. Баумана. Энергетическое и транспортное машиностроение. Специальный выпуск, 2011, с. 151-163.
 4. Low Chin Min Mike. Automatic targeting system for unmanned air vehicle operations. — National University of Singapore, 2006, 84 p.
 5. Abramov N.S., Makarov D.A., Khachumov M.V. Controlling flight vehicle spatial motion along a given route. – Automation and Remote Control, 2015, Vol.76, No.6, pp.1070-1080.
 6. Математическая модель приводов наведения видеокамеры в горизонтальной и вертикальной плоскостях. URL: http://knowledge.allbest.ru/radio/3c0b65625a2bc68a5d43a88421316c36_0.html (дата обращения: 21.03.2016).
 7. Абрамов Н.С., Хачумов М.В. Моделирование управления бортовой видеокамерой беспилотного летательного аппарата. – Авиакосмическое приборостроение, № 3, 2014, с.9-16.
 8. Абрамов Н.С., Ромакин В.А., Методы управления поворотной видеокамерой. – Известия ЮФУ. Технические науки, №7, 2013, с. 173–179.
 9. Хачумов М.В. Задача автоматического управления летательным аппаратом в процессе преследования цели. – Программные системы: теория и приложения, №5, 2014, с.67-77.

СЕКЦИЯ «ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И ПРИНЯТИЕ РЕШЕНИЙ»

ПРОБЛЕМЫ И РЕШЕНИЯ РАЗРАБОТКИ СРЕДСТВ АВТОМАТИЗАЦИИ ПОДДЕРЖКИ ДЕЯТЕЛЬНОСТИ ОПЕРАТОРА БУРОВОЙ УСТАНОВКИ

Абу-Абед Фарес Надимович

к.т.н., доцент кафедры ЭВМ ТвГТУ
ФГБОУ ВО «ТвГТУ»

г.Тверь, 170026, пр-т Комсомольский, д.5, ФМАС ТвГТУ, каб.513, aafares@mail.ru

Борисов Николай Анатольевич

к.т.н., доцент кафедры Программной Инженерии НИНГУ им. Н.И. Лобачевского
ФГБОУ ВО «ТвГТУ»

г.Нижний Новгород, 603950, пр. Гагарина, nborisov.y@yandex.ru

Аннотация

В статье сформулирован подход для разработки системы выявления предаварийных ситуаций на буровых установках. Для этого определена общая структура системы распознавания, сформирован априорный словарь признаков, используемых для распознавания предаварийных ситуаций при работе буровой установки. Разработан алгоритм определения рабочего словаря признаков на основе модифицированного метода ветвей и границ. Сформированы наборы признаков (рабочие словари) для распознавания наиболее распространенных предаварийных ситуаций на буровой.

Ключевые слова: Системы поддержки принятия решений; Распознавание образов; Нештатные (Предаварийные) ситуации; Признаки; Рабочие словари.

1. Введение

Внедрение в практику эксплуатации буровых установок средств анализа текущей обстановки на буровой и выявления нарушений в её работе позволяет снизить вероятность возникновения нештатных ситуаций за счёт использования специализированных средств поддержки принятия решений в работе бурового мастера [(Левицкий А.З., 2001), (Уолт, 2001), (Коваленко, 2002)]. Используемые средства могут различаться на концептуальном уровне, но в каждой системе есть специализированная база данных (база знаний), на основе которой принимаются решения и соответствующая программная надстройка, реализующая формальный алгоритм распознавания вида текущей ситуации.

Таким образом, для разработки средств поддержки деятельности оператора буровой были выделены следующие направления:

- Определение источника данных для оценки состояния буровой в процессе бурения;
- Выбора набора признаков, достаточного для распознавания текущего состояния буровой установки;
- Выбора способа построения и реализация программного классификатора, способного в динамике распознать предаварийную ситуацию и предупредить оператора буровой.

2. Определение источника данных для оценки состояния буровой в процессе бурения

Для оценки состояния буровой установки необходимо использовать данные, входящие в состав геолого-технологической информации (ГТИ), фиксируемые и сохраняемые информационно-измерительной системой буровой установки.

3. Формирование рабочего словаря признаков для распознавания нештатных ситуаций

В процессе распознавания возникает задача преобразования параметров распознаваемых объектов (входного набора признаков) в заключение об отношении распознаваемого объекта к одному из predetermined классов (результат распознавания) [(Абу-Абед Ф.Н., 2005)].

При этом желательно минимизировать количество признаков, используемых в процессе распознавания, но не в ущерб их информативности, которая должна гарантировать высокую достоверность получаемого результата.

Обозначим множество признаков объекта, поступающее на вход системы распознавания, как $\bar{P} = \{p_1, p_2, \dots, p_m\}$, $|\bar{P}| = m$. Результат распознавания будет представлять собой вектор-признак $\bar{X} = \{x_1, x_2, \dots, x_n\}$, $|\bar{X}| = n$, где $\bar{X} \subseteq \bar{P}$, $n \leq m$.

Определение набора признаков, которые представляют в системе распознавания существо распознаваемых явление или объектов, является первоочередной задачей при разработке систем такого класса. Исходной информацией для составления априорного словаря признаков может послужить используемая в данной предметной области нормативная документация, в случае бурения скважин – [ГОСТ Р 53375-2009]. На основании анализа его технических требований был сформирован априорный словарь, состоящий из 21 признака, используемых в настоящее время для технологических измерений и содержащих в себе необходимую информацию для распознавания нештатных ситуаций.

После формирования априорного словаря необходимо выполнить первоначальную классификацию объектов, которые подлежат распознаванию, т.е. составить априорный алфавит классов. В результате анализа существующих источников в качестве первоочередных ситуаций, подлежащих распознаванию выбраны:

- Газонефтеводопроявления;
- Поглощение бурового или тампонажного раствора;
- Прихваты бурового инструмента;
- Разрушение стенок скважины.

После определения априорного алфавита классов возможно формирование рабочего словаря признаков на основе априорного. При этом в качестве описания распознаваемого объекта будем использовать набор признаков $F = \{f1, \dots, fn\}$, каждый из которых имеет некоторое множество D_j допустимых значений, причем не обязательно числовое. Таким образом, вектор $f1(x), \dots, fn(x) \in D1 \times \dots \times Dn$ представляет собой признаковое описание объекта X .

Рабочий словарь признаков формируется посредством алгоритма, определяющего достаточный набор признаков для каждой из нештатных ситуаций, указанных ранее. В основу разработанного алгоритма положен метод ветвей и границ, модифицированный для сокращения области поиска путем ограничения его ширины на каждом шаге.

Критериальная функция, используемая на каждом шаге разработанного алгоритма, строится на основе использования коэффициентов корреляции признаков априорного словаря и признаков, обозначающих наличие или отсутствие той или иной нештатной ситуации (т.е. целевых признаков). Кроме этого, в качестве составляющей критериальной функции используются коэффициенты корреляции признаков между собой, что позволяет избежать необоснованного расширения признакового пространства.

Окончательно алгоритм формирования рабочего словаря признаков может быть сформулирован следующим образом:

Вход: множество F , критерий Q , параметр B ;

- 1: Инициализация массива лучших значений критерия: $Q_j^{\min} := +\infty$ для всех $j = 1, \dots, n$;
- 2: Упорядочить признаки по убыванию информативности;
- 3: Нарастить(\emptyset);
- 4: вернуть G , для которого $Q(G) = \min_{j=1, \dots, n} Q_j^{\min}$;
- 5: ПРОЦЕДУРА Нарастить (G);
- 6: если найдётся $j \leq |G|$ такое, что $Q(G) \geq Q_j^{\min}$, то
- 7: выход;
- 8: $Q_{|G|}^{\min} := \min\{Q_{|G|}^{\min}, Q(G)\}$;
- 9: для B лучших $f_i \in F$ таких, что $i > \max\{i | f_i \in G\}$;
Нарастить ($G \cup \{f_i\}$);

При добавлении признаков в рабочий набор на каждом шаге разработанного алгоритма из множества еще не выбранных признаков используются не более B (параметр алгоритма) признаков, имеющих наибольшее значение критерия.

В качестве критерия выбора признаков используется следующее выражение:

$$Q_N = \sum_{i \in T_N} K_i^H - \sum_{i \in T_N} \sum_{j \notin T_N} |K_{ij}|, \quad (1)$$

- где
- Q_N – значение критерия выбора признака на шаге N ;
 - T_N – множество уже выбранных признаков на шаге N ;
 - K_i^H – коэффициент корреляции признака i с целевым признаком;
 - K_{ij} – значение коэффициент корреляции признака i с признаком j .

Рисунки 1 и 2 демонстрируют результат работы алгоритма для предаварийной ситуации «Газонефтеводопроявления» при значениях $B = 3$ и $B = 4$.

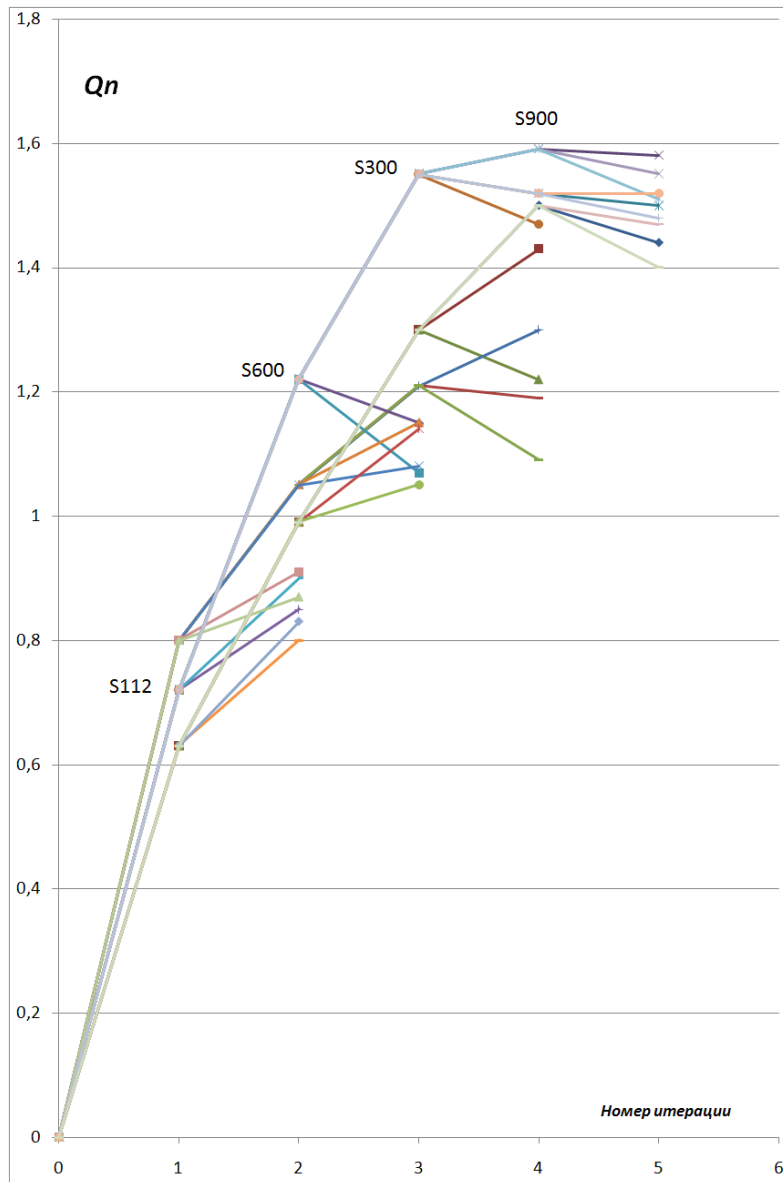


Рис 1. Результаты работы алгоритма выбора признаков для предаварийной «Газонефтеводопроявления» для ширины поиска $B = 3$

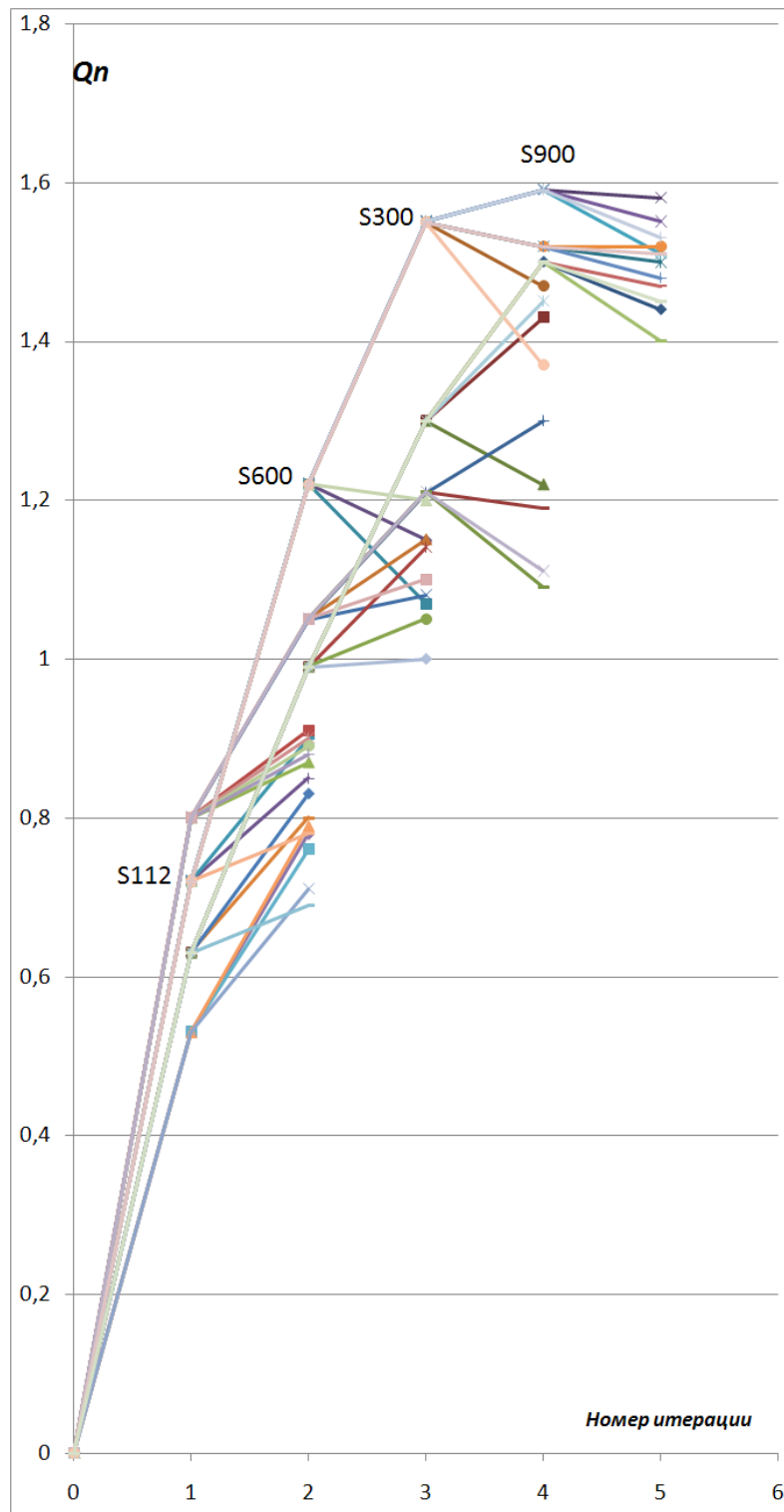


Рис 2. Результаты работы алгоритма выбора признаков для предаварийной «Газонефтеводопроявления» для ширины поиска $B = 4$

На данных рисунках демонстрируется работа алгоритма для одной и той же ситуации («Газонефтеводопроявления»), но с разными значениями параметра B . Например, при $B = 3$ (рисунок 1) из каждой вершины дерева порождаются еще три вершины (в рабочий словарь добавляются три новых признака, максимально коррелированные с наличием целевой ситуации).

На следующем шаге работы алгоритма из каждой полученной таким образом вершины дерева порождаются еще три, соответствующие признакам, дающим максимальное значение критерия (1), но минимально связанные с выбранным на предыдущем шаге признаком. Таким образом, для выбора двух признаков имеем девять вариантов, из которых для дальнейшего рассмотрения оставляем три наилучших и переходим к следующему шагу работы алгоритма.

Завершив пятый шаг работы, обнаруживаем, что найти признак, увеличивающий значение (1), в множестве еще не выбранных невозможно. Поэтому алгоритм заканчивает работу, а результатом будет являться множество вершин дерева (т.е. выбранных признаков), находящихся на ветви, которая заканчивается признаком S900, давшим при добавлении на четвертом шаге дал наибольшее значение критерия (1).

На этой ветви находятся (т.е. определены алгоритмом как наиболее информативные для распознавания ситуации «Газонефтеводопроявления») такие признаки:

- S112 (скорость бурения);
- S300 (давление на манифольде);
- S600 (плотность раствора на входе);
- S900 (температура раствора на выходе).

При этом признаки могут выбираться и использоваться в произвольном порядке, существенным является лишь состав полученного словаря.

Аналогичные результаты могут быть получены для той же ситуации при значении $B = 4$. При этом в каждой точке строящегося дерева порождается по 4 ветви, т.е. добавляется по 4 признака, дающих наилучшие значения (1). Для двух последовательных шагов получаем 16 точек, из них выбирается 4 наилучших и т.д. Результаты работы алгоритма представлен на рисунке 2, причем конечные результаты работы совпадают с полученными при значении $B = 3$, из чего можно сделать вывод, что использовать значение B большее трех нецелесообразно.

Аналогично описанному, были получены рабочие словари для распознавания всех вышеперечисленных нештатных ситуаций. При наличии в априорном словаре 21 признака для распознавания четырех типов нештатных ситуаций оказалось достаточным 9 признаков.

4. Построение нейросетевого классификатора для анализа ситуаций на буровой

Поскольку при распознавании предаварийных ситуаций возникающих в процессе бурения, размерность пространства признаков достаточно велика и границы между классами предаварийных ситуаций являются нечеткими, применение байесовской классификации и кластерного анализа для решения задачи распознавания предаварийных ситуаций в процессе проводки скважины представляется нецелесообразным, поэтому в качестве математического аппарата решения задачи распознавания предаварийных ситуаций (ПАС) предложено использовать искусственные нейронные сети прямого распространения, обучаемые с помощью метода (алгоритма) обратного распространения ошибки.

Математически процесс обучения нейросети описывается следующим образом: в процессе функционирования нейронная сеть формирует выходной сигнал Y в соответствии с входным сигналом X , реализуя некоторую функцию $Y = G(X)$. Если архитектура сети задана, то вид функции G определяется значениями синаптических весов и смещений сети. Пусть решением некоторой задачи является функция $Y = F(X)$, заданная парами входных - выходных данных $(X1, Y1), (X2, Y2), \dots, (XN, YN)$, для

которых $Y_k = F(X_k)$ ($k = 1, 2, \dots, N$). Обучение состоит в поиске (синтезе) функции G , близкой к F в смысле некоторой функции ошибки E . (рисунок 3):

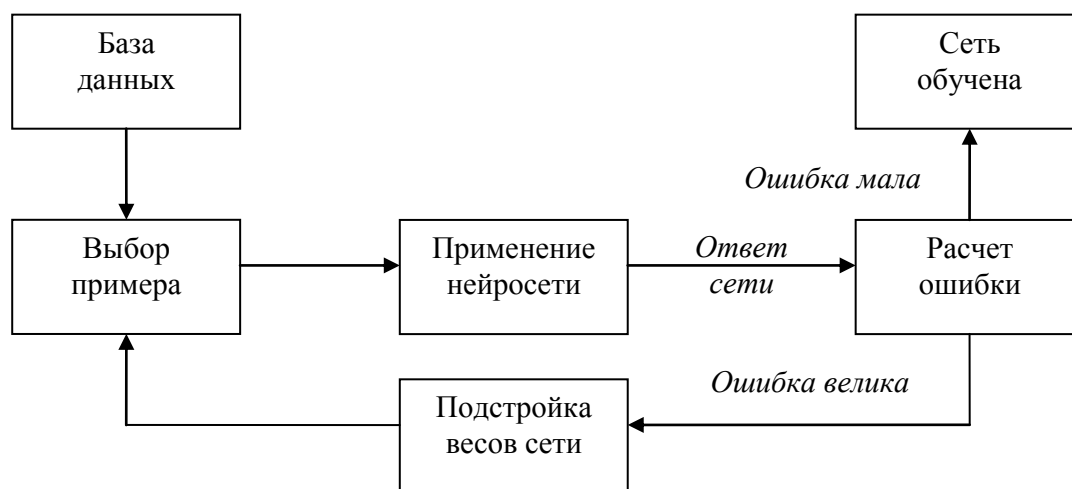


Рисунок 3. Иллюстрация процесса обучения НС

Если выбраны множество обучающих примеров – пар (X_k, Y_k) (где $k = 1, 2, \dots, N$) и способ вычисления функции ошибки E , то обучение нейронной сети превращается в задачу многомерной оптимизации, имеющую очень большую размерность, при этом, поскольку функция E может иметь произвольный вид, обучение в общем случае – многоэкстремальная невыпуклая задача оптимизации.

Для решения этой задачи могут быть использованы следующие алгоритмы: алгоритмы локальной оптимизации с вычислением частных производных первого порядка; алгоритмы локальной оптимизации с вычислением частных производных первого и второго порядка; стохастические алгоритмы оптимизации; алгоритмы глобальной оптимизации.

Алгоритм обратного распространения ошибки применяется для обучения многослойных нейронных сетей с последовательными связями, на основе которых построен классификатор предаварийных ситуаций. Это итеративный градиентный алгоритм, который используется с целью минимизации среднеквадратичного отклонения текущего выхода многослойного персептрона и желаемого выхода.

Согласно методу наименьших квадратов, минимизируемой целевой функцией

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2,$$

ошибки НС, является величина:

где $y_{j,p}^{(N)}$ – реальное выходное состояние нейрона j выходного слоя N нейронной сети при подаче на ее входы p -го образа; $d_{j,p}$ – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}},$$

образом:

Здесь w_{ij} – весовой коэффициент синаптической связи, соединяющей i -ый нейрон слоя $n-1$ с j -ым нейроном слоя n ; η – коэффициент скорости обучения, $0 < \eta < 1$.

Существующие методы улучшения качества обучения с помощью данного алгоритма связаны с использованием различных функций ошибки, процедур

определения направления и величины шага, процедур составления расписания обучения и других.

В качестве основного недостатка этих методов можно отметить отсутствие в них эффективной процедуры определения глобальности минимума целевой функции. В работе предложена модификация базового алгоритма обучения, ориентированная на нахождение этого глобального минимума. Ее суть заключается в следующем:

- После достижения локального минимума (завершение процесса обучения), случайным образом генерируется новая начальная точка, и процесс обучения повторяется;
- После достижения (с заданной точностью E) K раз из разных начальных точек одного и того же минимального значения ошибки обучения (K – параметр алгоритма) меняется параметр R генерации начальных точек для процесса обучения (радиус поиска увеличивается), и процесс поиска глобального экстремума продолжается (заново производится обучение нейросети);
- После достижения параметром R значения R_{max} (R_{max} – параметр алгоритма) нейросеть считается обученной, и алгоритм заканчивает работу.

Блок-схема предложенного алгоритма приведена на рисунке 4.

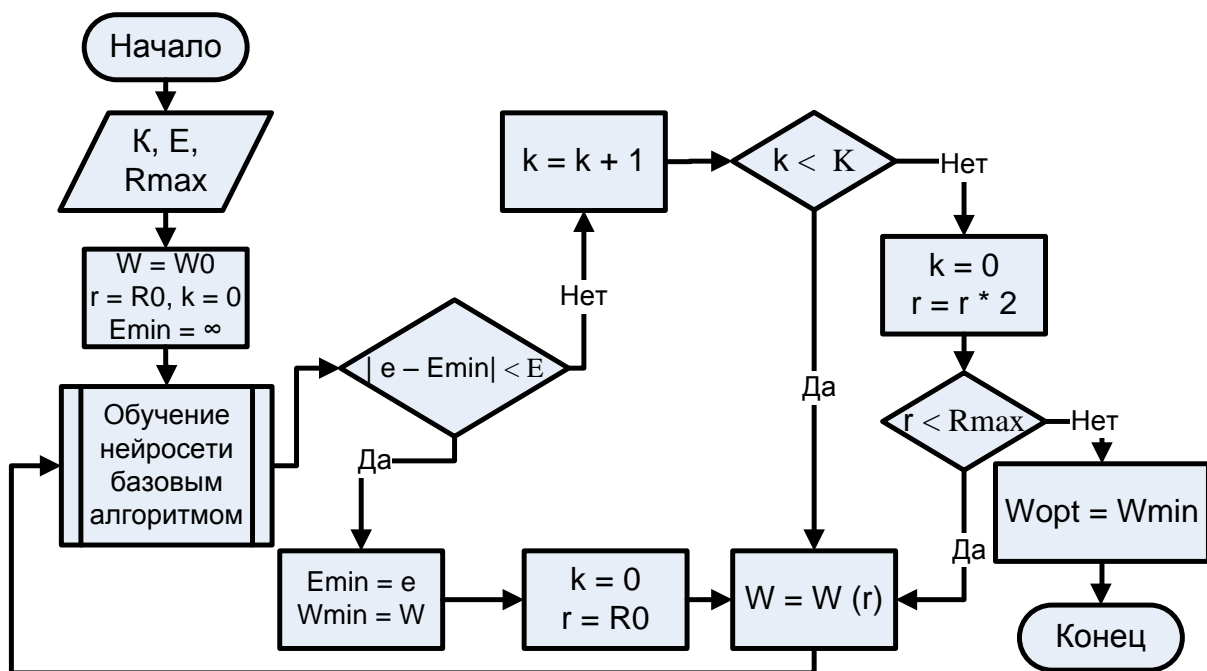


Рис 4. Блок-схема модифицированного алгоритма

Классификатор для распознавания предаварийных ситуаций на основе искусственной нейронной сети может быть реализован двумя способами:

- С формированием на выходе N различных сигналов, каждый из которых соответствует одной распознаваемой ситуации, которые формируются на основе анализа M признаков, т.е. с использованием единого словаря признаков для всех распознаваемых ситуаций.
- Как состоящий из N классификаторов, каждый из которых способен распознать одну предаварийную ситуацию и имеет один выход и m_i входов, причём множества признаков, используемых для распознавания различных ПАС, могут перекрываться.

Преимуществом первого подхода является достаточность разработки и обучения одного классификатора для всего множества распознаваемых ситуаций. Однако при этом затраты времени на обучение такого классификатора могут оказаться больше, чем на обучение нескольких классификаторов меньшего размера.

С другой стороны, точность распознавания у классификаторов, ориентированных на конкретную ПАС, может оказаться лучше, чем у одного универсального классификатора.

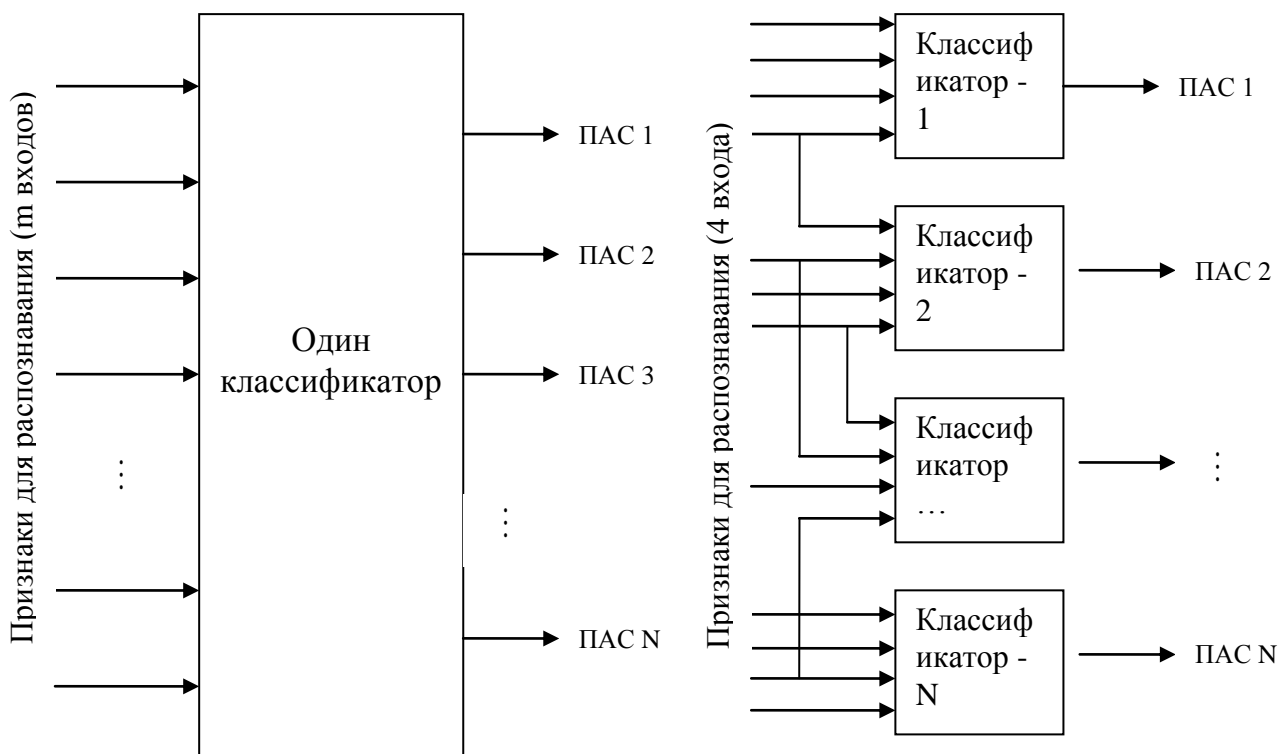
Окончательное принятие решения по выбору структуры классификатора возможно только на основе исследований эффективности обоих вариантов.

Для обеих представленных структур классификатора были проведены исследования влияния числа слоев нейросети и количества нейронов в каждом скрытом слое на:

- Количество необходимых итераций обучения нейросети;
- Точность распознавания обученной нейросетью векторов обучающей выборки.

На рисунке 5 представлены оба варианта построения классификатора.

Для определения структуры нейросетевого классификатора ПАС проведены экспериментальные исследования влияния параметров нейросети на эффективность обучения распознаванию искусственно сгенерированных ситуаций. На рисунках 6 и 7 приведены полученные графики для этих зависимостей.



а) Универсальный классификатор для всех ПАС

б) Специализированные классификаторы для каждой ПАС

Рис. 5. Варианты построения нейросетевого классификатора

График количества итераций обучения для разного числа слоёв и числа нейронов скрытого слоя

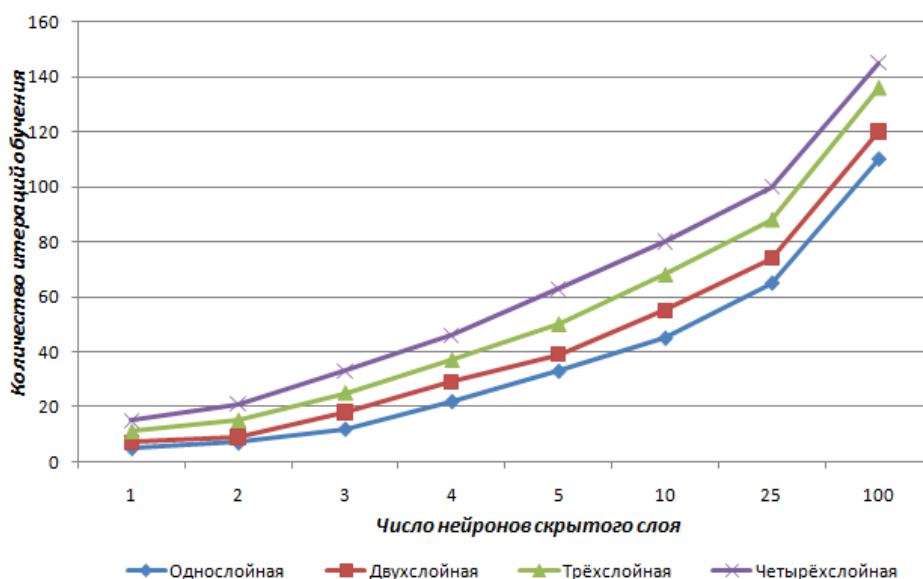


Рис 6. Графики Зависимость числа итераций обучения от числа слоёв и количества нейронов в каждом слое

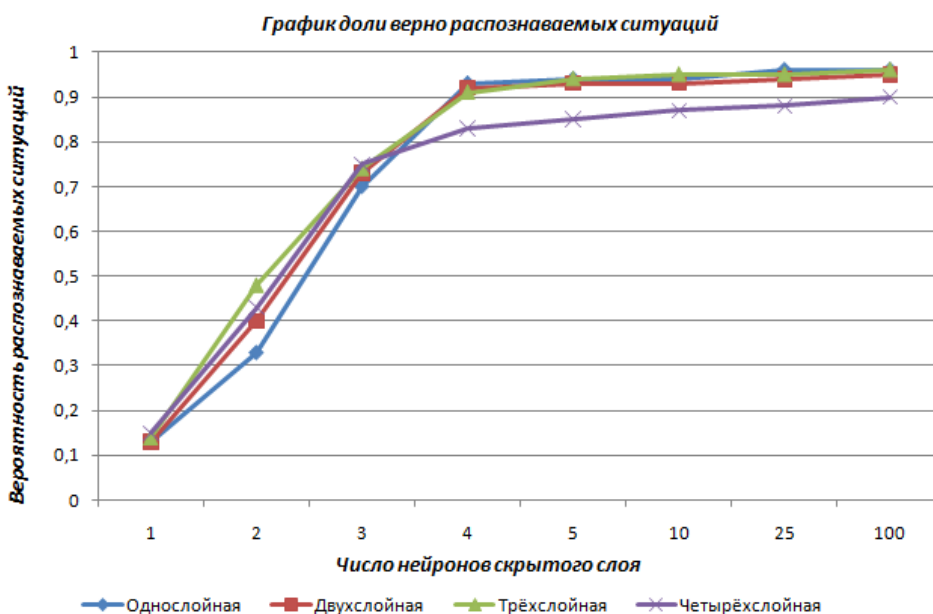


Рис. 7. Зависимость точности распознавания от числа слоёв и количества нейронов в каждом слое

Для использования на практике выбрана структура специализированного нейросетевого классификатора, состоящая из одного скрытого слоя с числом нейронов, равным числу входов классификатора.

Результаты проведённых исследований позволяют сформулировать метод распознавания предаварийных ситуаций в процессе бурения нефтяных и газовых скважин с помощью нейросетевого классификатора как систематизированную последовательность действий:

1. Определение набора признаков, доступных для измерения существующими средствами обработки ГТИ, и формирование априорного словаря признаков.

2. Классификация состояний объекта с целью определения совокупности ПАС, подлежащих распознаванию.

3. Формирование рабочих словарей признаков для каждой распознаваемой ПАС с помощью разработанного алгоритма на основе метода ветвей и границ.

4. Определение структуры классификатора ПАС, построенного на основе нейросети прямого распространения, обучаемой модифицированным методом обратного распространения ошибки.

5. Обучение разработанного классификатора ПАС на реальных данных исследуемой предметной области.

5. Заключение

В работе представлен алгоритм для формирования рабочего словаря признаков, позволяющий выбрать признаки, необходимые для четкого распознавания наиболее распространенных нештатных ситуаций, возникающих в процессе бурения нефтяных и газовых скважин, из множества измеряемых технологических параметров. Сформированные рабочие словари использованы при разработке комплекса программ, предназначенных для распознавания нештатных ситуаций на буровой установке [(Абу-Абед Ф.Н., 2009), (Абу-Абед Ф.Н., 2006)].

Проведен анализ основных математических методов, применяемых для решения задачи распознавания образов. В качестве математического аппарата для решения задачи определения текущего состояния объекта исследования по заданному набору признаков выбран метод распознавания образов на основе искусственной нейронной сети. Разработан модифицированный алгоритм обучения классификатора ПАС на основе метода обратного распространения ошибки, который отличается от классического наличием процедуры поиска глобального минимума функции ошибки, и осуществлена его программная реализация.

Предложена общая структура нейросетевого классификатора ПАС, показана возможность и целесообразность решения задачи распознавания для каждой ПАС в отдельности, для чего выполнена декомпозиция задачи построения нейросетевого классификатора. Разработана структура нейросетевого классификатора, состоящая из одного скрытого слоя с числом нейронов, равным числу входов классификатора. Полученные результаты сведены в обобщенный метод распознавания ПАС в процессе промышленного бурения нефтяных и газовых скважин.

Литература

1. Абу-Абед Ф.Н. Разработка средств моделирования нейросетей. Вестник Тверского государственного технического университета. 2005. № 7. С. 125-129.
2. Абу-Абед Ф.Н. Надежность объектов нефтегазодобывающей промышленности. Газовая промышленность. 2015. № S720 (720). С. 107-111.
3. Абу-Абед Ф.Н., Допира Р.В. Применение средств моделирования нейросетей для анализа предаварийных ситуаций на буровых. Программные продукты и системы. 2010. № 3. С. 136-139.
4. Абу-Абед Ф.Н. Автоматизированная система обнаружения предаварийных ситуаций на объектах нефтегазодобывающей промышленности. Каротажник. 2015. № 5 (251). С. 48-61.
5. Абу-Абед Ф.Н. Снижение риска при строительстве газовых скважин на базе нейросетевой модели. Газовая промышленность. 2014. № S712 (712). С. 100-102.
6. Абу-Абед Ф.Н. Борисов Н.А., Хабаров А.Р. Использование методов распознавания образов для анализа аварийных ситуаций [Конференция] // Математические методы

- и информационные технологии в экономике, социологии и образовании: сборник статей XVI Межд. НТК.. - Пенза : ПДНТП, 2005. - стр. 428-431.
7. Абу-Абед Ф.Н. Хабаров А.Р. Применение нейросетей для анализа аварийных ситуаций на буровых [Конференция] // Математические методы и информационные технологии в экономике, социологии и образовании: сборник статей XVII Межд. НТК.. - Пенза : ПДНТП, 2006. - стр. 218-221.
 8. Абу-Абед Ф.Н. Программа построения и обучения нейросети для распознавания режимов работы буровой. // Свидетельство о регистрации программы для ЭВМ № 2009615089. - М. : Роспатент, 2009 г.
 9. Левицкий А.З. Тенишев В.М., Командровский В.Г. Компьютерные и информационные технологии в решении задач оперативного управления бурением. Ч. 3. Информационные и компьютерные технологии в обработке первичных данных с целью предупреждения, ранней диагностики и ликвидации осложнений. - Москва : РГУ нефти и газа имени И. М. Губкина, 2001 г.
 10. Р.А. Коваленко Программно-аппаратные средства информационно-измерительной системы управления процессом бурения скважин : диссертация ... кандидата технических наук : 05.11.16, 05.13.18. - Ижевск : Ижевский государственный технический университет, 2002 г..
 11. Уолт Алдрид Управление риском в бурении [Статья] // Нефтегазовое обозрение. - Москва : ООО "Бонаэкспо", 2001 г.. - 1 : Т. 6. - стр. 12-29.

МЕТОДЫ ИДЕНТИФИКАЦИИ БЛОКИРУЮЩЕГО ПРЕПЯТСТВИЯ В ЗАДАЧАХ ПЛАНИРОВАНИЯ ТРАЕКТОРИИ

Андрейчук А.А.
РУДН
andreychuk@mail.com
Яковлев К.С.
ФИЦ ИУ РАН
yakovlev@isa.ru

Аннотация

Задача планирования траектории обычно рассматривается в искусственном интеллекте как задача поиска пути на графе особой структуры, и все известные алгоритмы предполагают получение в качестве ответа путь, если таковой существует, или же специального ответа «пути не существует» в противном случае. При этом в ряде задач, например при планировании совокупности траекторий для множества интеллектуальных агентов, обладающих различными возможностями взаимодействия с окружающей средой, необходимо знать *почему* пути не существует. Если, это обусловлено наличием блокирующего препятствия, то последнее может быть разрушено одним из участником коалиции агентов и ранее не решаемая задача планирования теперь может иметь ответ. Таким образом проблема идентификации блокирующего препятствия при планировании становится актуальной и востребованной. Исследованию и разработке методов решения этой проблемы и посвящена настоящая работа.

1. Введение

Задача планирования траектории в искусственном интеллекте обычно рассматривается как задача поиска пути на взвешенном графе, вершинам которого соответствуют положения агента в пространстве, а ребрам – возможные переходы между ними. При этом весьма распространено использование графа специального вида – метрического топологического графа (в англоязычной терминологии – grid) [1], который является простой и вместе с тем информативной графовой моделью окружающей агента среды. Метрический топологический граф (МТ-граф) – взвешенный неориентированный граф, каждый элемент (вершина) которого соответствует некоторой области пространства и может быть либо проходим, либо непроходим для агента. При планировании траектории на плоскости, МТ-граф можно представить в виде таблицы (см. Рис.1).

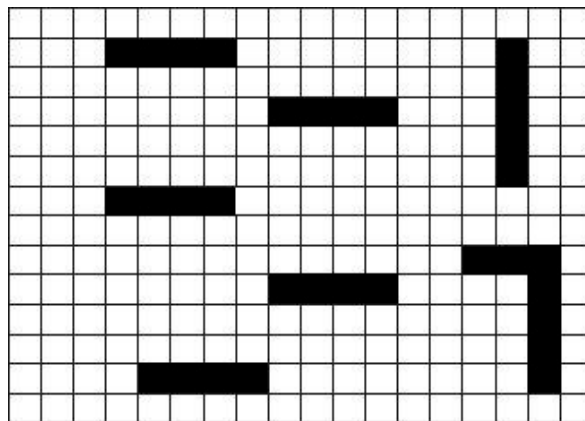


Рис.1– Пример графического представления МТ-графа

Формально МТ-граф – это тройка: $Gr = \langle A, ADJ, d \rangle$, где A – множество вершин, представляющее собой матрицу $A_{m \times n} = \{a_{ij}\}: a_{ij} = 0 \vee a_{ij} = 1, \forall i, j: 0 \leq i < m, 0 \leq j < n, m, n \in N$; $ADJ \subseteq A \times A$ – отношение, задающее смежность на множестве вершин (множество смежных вершин); d – метрика на множестве A (функция, задающая способ определения расстояния между вершинами).

Вершины $a_{ij} = 0$ будем считать проходимыми, а $a_{ij} = 1$ – непроходимыми. Вершины a_{ij} , a_{lk} будем называть смежными, если: $\forall i, l, j, k: 0 \leq i, l < m, 0 \leq j, k < n \rightarrow |i - l| \leq 1 \wedge |j - k| \leq 1$. Множество вершин, смежных с a_{ij} будем обозначать как $adj(a_{ij})$. В качестве метрики на МТ-графе будем использовать Евклидову метрику: $d(a_{ij}, a_{lk}) = \sqrt{(i - l)^2 + (j - k)^2}$

Пусть зафиксированы две вершины графа: начальная – s , и целевая – g . Путем из s в g будем считать последовательность смежных, проходимых вершин $\pi(s, g) = \{a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_k j_k}\}$, таких, что $a_{i_v j_v} \in adj(a_{i_{v+1} j_{v+1}}), v = \overline{1, k-1}, a_{i_1 j_1} = s, a_{i_k j_k} = g$. Длина пути равна сумме длин переходов между вершинами, составляющими путь $d(\pi(s, g)) = \sum_{v = \overline{1, k-1}} d(a_{i_v j_v}, a_{i_{v+1} j_{v+1}})$.

Задача планирования траектории представляется теперь в виде тройки:

$$PTask = \langle Gr, s, g \rangle$$

и формулируется следующим образом: пусть задан МТ-граф Gr и зафиксированы две вершины: начальная – s , и целевая – g , необходимо построить путь $\pi(s, g)$.

Одним из наиболее распространенных в искусственном интеллекте алгоритмов, решающих поставленную задачу, является алгоритм A^* [2], а также его различные модификации. Например, JumpPointSearch[3] – алгоритм, нацеленный на улучшение временной и емкостной эффективности, или Theta*[4] – его целью является

уменьшение длинотыскиваемых путей. Перечисленные алгоритмы являются полными и корректными, то есть гарантируют, что решение будет найдено, если оно существует, либо, если решения не существуют – сообщат об этом. При этом во многих практических задачах, например, при навигации беспилотных транспортных средств, возникает потребность в более информативном ответе при невозможности построить путь. Для системы управления, частью которой является планировщик, важно *почему* путь до цели построить невозможно. Очевидно, что это происходит из-за наличия некоторого препятствия, идентификация которого становится весьма актуальной задачей. Разработке процедуры идентификации блокирующего препятствия, а также модификации алгоритмов эвристического поиска A*, JPS и Theta*, и посвящена данная работа.

2. Алгоритм A* с идентификацией блокирующего препятствия

В основе алгоритма A* лежит процесс итерационного обхода графа и расчёта для каждой рассматриваемой вершины графа некоторого числового значения – *g*-значения. $g(a)$ – это длина пути от начальной вершины до вершины a , вычисленная к текущей итерации алгоритма. Перед началом работы алгоритма полагается, что $g(a) = \infty$ для всех вершин, кроме начальной ($g(s) = 0$). В ходе работы алгоритма *g*-значения вершин уменьшаются в результате процедуры раскрытия (от англ. expand). Под раскрытием вершины a подразумевается процесс исследования всех смежных вершин $adj(a)$ на предмет возможности уменьшения их *g*-значений. В процессе раскрытия для каждой смежной вершины $a' \in adj(a)$ рассчитывается её *g*-значение. Если $g(a') > g(a) + d(a, a')$, то значение $g(a')$ обновляется. Помимо *g* значений, для каждой смежной вершины a' в памяти сохраняется её родитель – текущая вершина a .

Все вершины, которые были рассмотрены алгоритмом, хранятся в двух списках – *open* и *close*. Изначально список *close* пуст, а в списке *open* находится начальная вершина. На очередной итерации алгоритма, после того, как текущая вершина a раскрывается, она добавляется в список *close*. А каждая смежная вершина $a' \in adj(a)$ добавляется в список *open*, при условии, что этой же вершины нет в списке *close*. Если вершина попала в список *close*, то это значит, что путь минимальной длины до неё уже найден.

На каждой итерации алгоритм раскрывает одну вершину из списка *open* с минимальным значением $f(a) = g(a) + h(a)$, где $h(a)$ – это эвристическая оценка длины пути до целевой вершины, вычисляемая с помощью метрики МТ-графа. Алгоритм A* работает до тех пор, пока целевая вершина не окажется в списке *close*. После этого алгоритм восстанавливает путь по обратному указателю на родительскую вершину.

В случае, когда пути не существует, целевая вершина никогда не будет достигнута и соответственно не будет добавлена в список *open* или *close*. В таком случае алгоритм исследует все доступное пространство поиска и остановится, когда список *open* опустеет. После того, как алгоритм завершит свою работу, в списке *close* будут находиться все рассмотренные вершины (рис. 2).



Рис.2 – Черным цветом отмечены непроходимые области – препятствия, серым – области, которые были исследованы алгоритмом в процессе поиска, белым – области, недостижимые из начальной вершины

Если алгоритм завершил свою работу и не смог построить путь, значит, есть блокирующее препятствие. При этом, их может быть несколько, вложенных друг в друга, однако в данной работе рассматривается проблема идентификация внешнего блокирующего препятствия.

Предлагаемый метод идентификации блокирующего препятствия, основан на следующем наблюдении. После завершения работы алгоритма A^* , в списке `close` содержится информация о том, какие вершины были рассмотрены в процессе поиска, а какие нет. Более того, для каждой из рассмотренных вершин известны их g и h значения. Вершины, которые граничат с блокирующим препятствием, имеют относительно низкие h значения, так как находятся максимально близко к целевой вершине. Нетрудно убедиться, что вершина с минимальным h значением всегда смежна с блокирующим препятствием (Рис.3). Будем называть такую вершину пограничной.

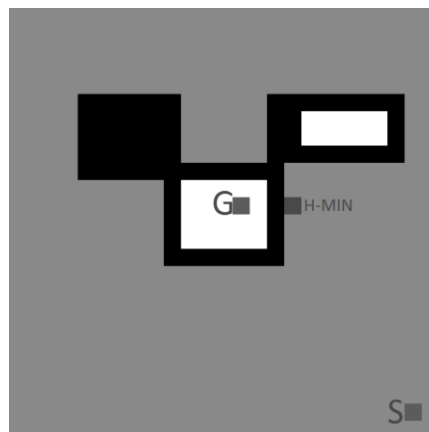


Рис.3 – Вершина с минимальным h значением всегда смежна с блокирующим препятствием

Заметим, что пограничная вершина может быть смежна с несколькими препятствиями, но зная координаты этой вершины и координаты цели, можно определить их положение относительно друг друга. Очевидно, что препятствие, которое блокирует проход к цели, всегда находится между пограничной и целевой вершиной. То есть, если целевая вершина находится выше пограничной, то и блокирующее препятствие находится выше.

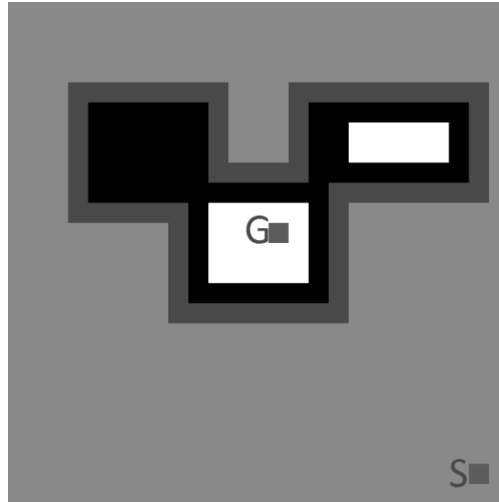


Рис.4 – Контур блокирующего препятствия

После того, как была найдена пограничная вершина и определено её положение, простым итерационным обходом можно составить контур блокирующего препятствия (Рис.4). После того, как контур составлен, можно сказать, что блокирующее препятствие найдено.

3. **Theta* и JumpPointSearchs идентификацией блокирующего препятствия**

Алгоритм Theta* работает по такому же принципу, что и алгоритм A*. Разница заключается лишь в том, что алгоритм Theta* может строить пути, которые не привязаны к топологии МТ-графа, то есть переход из текущей вершины разрешен не только в одну из восьми смежных вершин, но и в любую другую вершину, в которую можно перейти по прямой, не пересекая при этом препятствия. Такой принцип перехода даёт возможность строить более короткие пути с меньшим количеством поворотов. Достигается это за счет введения дополнительной проверки на существование проходимой прямой между текущей вершиной и родителем родителя текущей вершины. Как правило, для построения дискретной прямой используется алгоритм Брезенхема[5]. Если такая прямая существует, то родитель родителя становится родителем текущей вершины. Подобные проверки никак не сказываются на основном итерационном принципе работы алгоритма и не влияют на исследуемое пространство поиска. В связи с этим, предложенный выше метод идентификации блокирующего препятствия применим и для алгоритма Theta*.

Принцип работы алгоритма JumpPointSearchs существенно отличается от принципа работы алгоритма A* или Theta*. Основная цель алгоритма JPS – повышение вычислительной эффективности по сравнению с A*. Ускорение поиска происходит за счет того, что алгоритм по определенным правилам запоминает и раскрывает только некоторые

вершины, называемые опорными. Из-за этого в списке closeхранится не всё пространство поиска, а лишь небольшая его часть, состоящая из опорных вершин (Рис.5).



Рис.5 – Черным цветом отмечены непроходимые области – препятствия, серым – опорные вершины, которые алгоритм сохранил в памяти

Принцип работы алгоритма JPS делает невозможным применение метода идентификации блокирующего препятствия, предложенного ранее для алгоритмов A^* и $Theta^*$. Связано это с тем, что, во-первых, пограничная вершина с минимальным h значением зачастую не граничит с блокирующим препятствием. Во-вторых, не всегда можно определить, где находится блокирующее препятствие, основываясь на координатах целевой и пограничной вершины с минимальным h значением.

В связи с этим предлагается следующая модификация принципа работы алгоритма JPS. Необходимо рассчитывать h -значение для каждой вершины, которую рассматривает JPS в процессе поиска, включая те, что не являются опорными (jump points в терминологии оригинального алгоритма). При этом необходимо и достаточно хранить только одну вершину, чье h значение минимально, проверяя и при необходимости обновляя её значения в процессе работы алгоритма. В итоге, после завершения работы алгоритма будет найдена пограничная вершина. Подобная модификация алгоритма JPS позволяет использовать предложенный ранее метод идентификации блокирующего препятствия.

4. Экспериментальные исследования

Для проведения экспериментальных исследований, основываясь на оригинальных псевдокодах, были программно реализованы алгоритмы A^* , $Theta^*$ и JumpPointSearch. После чего алгоритмы были соответствующим образом модифицированы (см. предыдущий раздел) с целью возможности идентифицировать блокирующие препятствия, используя предложенный метод. Во всех трех алгоритмах использовались идентичные структуры хранения данных. В качестве языка программирования использовался язык C++11. Компьютер, на котором проводились экспериментальные исследо-

вания, имел следующую конфигурацию: OS–Windows7, CPU–IntelQ8300 (2.5GHz), RAM – 2 GB.

Алгоритмы были протестированы на коллекции карт реальной городской местности(Москвы), состоящей из 100 карт. Каждая карта представляет собой МТ-граф размером 501x501 вершин, с плотностью заполнения 20-25%. Для каждого графа было сформировано по 10 заданий, при этом начальная и целевая вершина располагались на противоположных концах карты.

Чтобы протестировать алгоритмы на заданиях, где пути не существует, карты были модифицированы следующими способами:

- 1) Ровно посередине карты проведена непроходимая линия. В зависимости от взаимного расположения стартовой и целевой вершины линия проведена либо горизонтально, либо вертикально. Таким образом, доступная область поиска составляет примерно 50% от всего размера карты.
- 2) Непроходимой линией огорожена четверть карты, причем таким образом, чтобы в эту четверть входила целевая вершина. Доступная область поиска составляет около 75%.
- 3) Непроходимой линией огорожена область размером 100x100 вершин. При этом целевая вершина находится в центре непроходимой области. Доступная область поиска составляет около 96%.
- 4) Непроходимой линией огорожена область размером 50x50 вершин. Доступная область поиска составляет около 99%.
- 5) Непроходимой линией огорожена область размером 10x10 вершин. Доступная область поиска составляет 99,96%.

Таким образом, была проведена серия экспериментов на 1000 заданиях, каждое из которых было протестировано на 6 картах, 5 из которых были модифицированы так, чтобы пути не существовало.

Эффективность работы алгоритмов оценивалась по двум критериям:

- 1) Память – измеряется в количестве вершин, которое потребовалось сохранить в памяти, чтобы найти путь, либо определить, что пути не существует.
- 2) Время – время работы непосредственно самого алгоритма поиска, без учета накладных расходов, таких как загрузка карты или сохранение результата. Измеряется в секундах.

Далее приведены результаты тестирования алгоритмов JPS, A* и Theta* на не модифицированных картах.

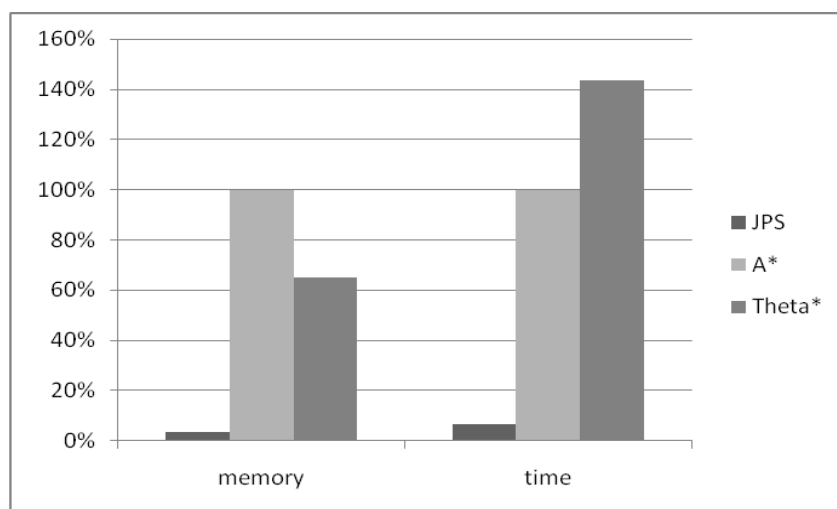


Рис. 6 Средние значения затрат времени и памяти, нормированные по алгоритму A*

Как видно из графика на Рис.6, алгоритм JPS работает примерно в 15 раз быстрее и потребляет в 30 раз меньше памяти, по сравнению с алгоритмом A*. Алгоритм Theta* потребляет меньше памяти, но при этом работает дольше. Объяснить это можно тем, что временная стоимость одной итерации Theta* больше, чем у A*, так как Theta* пытается сменить родителя текущей вершины, делая при этом дополнительные проверки.

Далее приведены результаты экспериментов на всех 5 видах модифицированных карт.

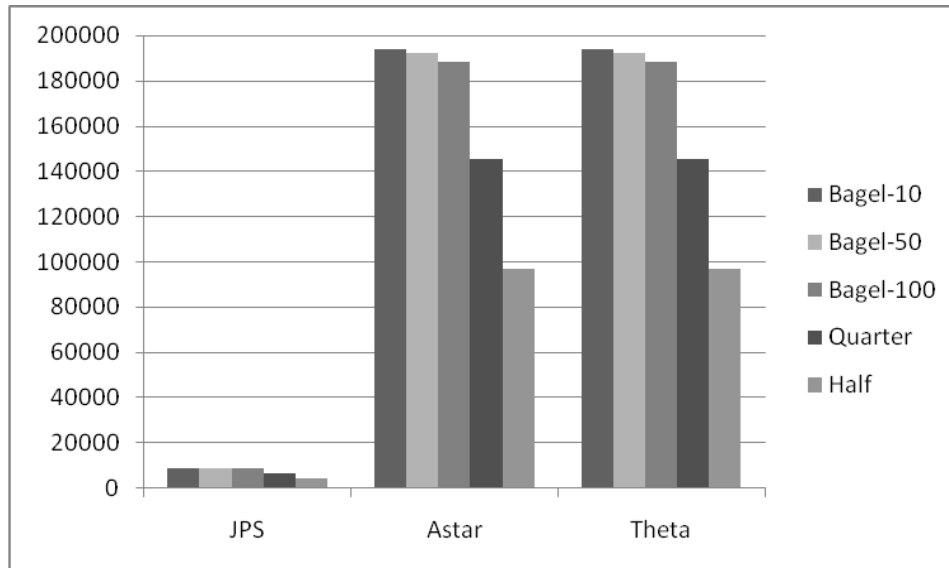


Рис.7 – Затраты памяти на всех 5 типах модифицированных карт

Показатели затрат памяти прямо пропорционально зависят от размера области поиска, доступной алгоритму для просмотра. Полученные значения для алгоритмов A* и Theta* являются абсолютно идентичными. Объяснить это можно тем, что принцип их работы, связанный с раскрытием и запоминанием вершин идентичный. В то время, как у алгоритма JPS он другой, в связи с чем его затраты во всех случаях существенно ниже.

На Рис.8 представлены результаты того же эксперимента, но связанные уже с временным показателем.

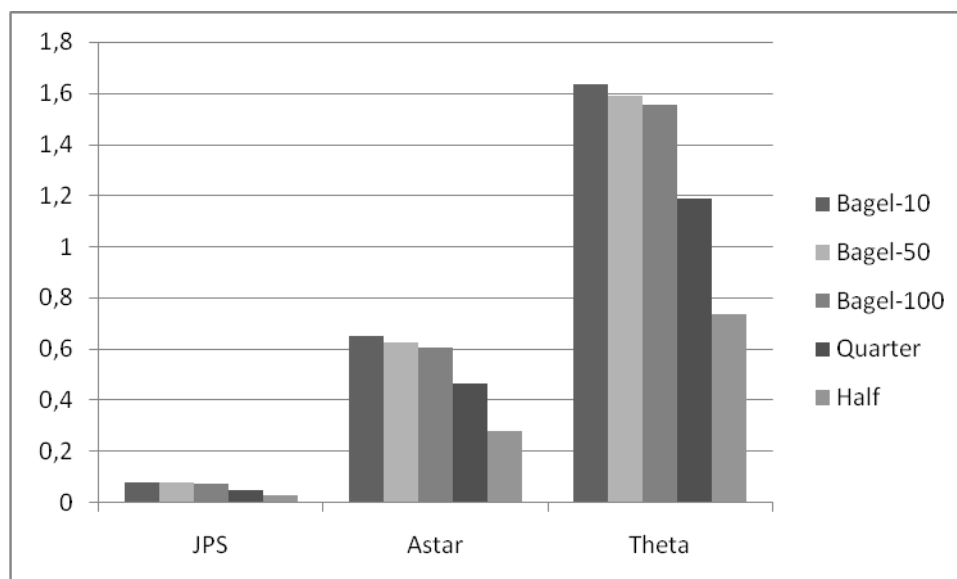


Рис.8–Временные затраты на всех 5 типах модифицированных карт

Как видно из графика на Рис.8 алгоритм JPS работает быстрее A* примерно в 9 раз, что несколько меньше, чем 15-кратное преимущество, которое демонстрировал JPS на тех заданиях, где путь существует. Алгоритм Theta* работает значительно медленнее алгоритма A*. В заданиях, где путь существует, Theta* работает в среднем на 40% медленнее, а в случае наличия блокирующего препятствия, алгоритм Theta* работает в 2,5 раза медленнее, так как фактически просматривает абсолютно такое же количество вершин, но при этом выполняет дополнительную работу, связанную со сменой родителя.

Основываясь на полученных результатах, можно сделать вывод, что наиболее сложными заданиями являются те, где цель заблокирована с помощью препятствия 10x10. На Рис.9 представлен график, в котором сравниваются показатели времени и памяти между случаями, когда путь существует и когда цель заблокирована в области 10x10.

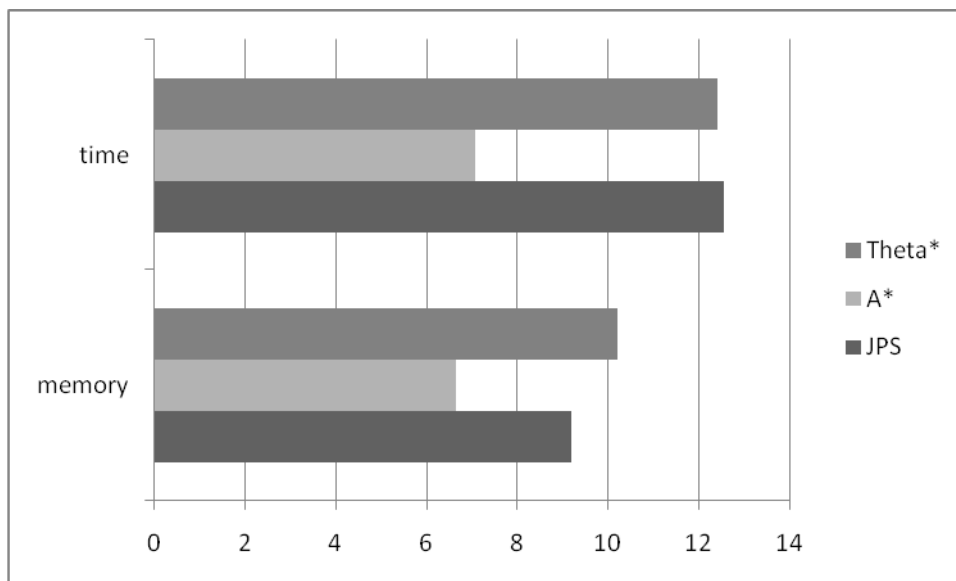


Рис. 9—Разница в показателях памяти и времени между случаями, когда путь существует и когда цель заблокирована в области 10x10

Данные, представленные на Рис.9, говорят о том, что решение заданий, где пути не существует, требует на порядок больше времени и памяти, чем решение заданий, где путь существует.

5. Заключение

В работе рассмотрена задача планирования траектории в тех случаях, когда пути не существует. Был предложен метод, который позволяет идентифицировать блокирующее препятствие. На основе этого метода были предложены модификации известных алгоритмов, используемых при планировании траектории на плоскости: A*, Theta* и JumpPointSearch. Проведены соответствующие экспериментальные исследования. Полученные в ходе экспериментов результаты говорят о том, что на определение того, что пути не существует, требуется на порядок больше памяти и времени. Сравнительный анализ показал, что несмотря на необходимость дополнительной модификации алгоритма JPS, замедляющей его работу, он остается на порядок более эффективным алгоритмом с точки зрения затрат как по памяти, так и по времени, по сравнению с алгоритмами A* и Theta*.

Литература

1. Яковлев К.С. Графы специальной структуры в задачах планирования траектории. Труды III международной конференции «Системный анализ и информационные технологии САИТ-2009». М: ИСА РАН, 2009.
2. P. E. Hart, N. J. Nilsson, B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics SSC4. – 1968. – № 2. P. 100 – 107.
3. D. Harabor, A. Grastien. "Online Graph Pruning for Pathfinding on Grid Maps". 25th National Conference on Artificial Intelligence. AAAI – 2011.
4. A. Nash, K. Daniel, S. Koenig & A. Felner, 2007. Theta*: Any-Angle Path Planning on Grids. In Proceedings of the National Conference on Artificial Intelligence (Vol. 22, No. 2, p. 1177). Menlo Park, Calif.: AAAI Press.
5. Bresenham, J. E. 1965. Algorithm for computer control of a digital plotter. IBM Systems journal, 4(1), 25-30.

РАЗРАБОТКА МОДУЛЯ ЗАДАНИЯ ПРАВИЛ ПРОДУКЦИОННОЙ СИСТЕМЫ

А.С.Горященко

Федеральный исследовательский центр «Информатика и управление» РАН,
Институт системного анализа
117312, г.Москва, проспект 60-летия Октября, д.9
e-mail: a.goriashchenko@gmail.com

Аннотация

В работе предлагается подход к созданию модуля задания правил для продукционной интеллектуальной системы. Наличие такого модуля в составе продукционной системы позволяет иметь простой и удобный механизм задания правил, который важен для ее разработки, для упрощения работы инженеров по знаниям и экспертов в предметной области.

1. Введение

Интеллектуальные системы с успехом используются в различных областях, начиная от систем поддержки принятия решения по назначению медицинских препаратов больным в зависимости от их состояния и симптомов заболевания, систем моделирования бизнес-процессов, систем анализа электронных документов, и заканчивая системами по управлению в реальном времени группами независимых автономных устройств, производственными циклами предприятий [1]. Одним из видов интеллектуальных систем являются продукционные системы или системы, основанные на правилах. Несмотря на то, что известно много работ по разработке автоматизированных методов приобретения знаний интеллектуальными системами [2], по-прежнему важную роль играет получение информации о предметной области от

эксперта-специалиста в этой области. В большинстве случаев эксперт не является программистом – специалистом по знаниям, поэтому для эффективной работы эксперта ему требуется содействие инженера по знаниям. Инженер по знаниям - программист, специалист в области искусственного интеллекта, выступает в роли промежуточного звена между экспертом и базой знаний. Для эффективной работы инженера по знаниям требуется предоставить удобный ему механизм задания правил для описания предметной области.

2. Правила в продукционных системах

Продукционная система состоит из рабочей памяти, множества правил, выполняющих некоторые действия (во внешней среде и в рабочей памяти) и некоторой стратегии управления [1]. Правила применяются к рабочей памяти. Правилем будем называть предложение типа «Если (условие), то (действие)». Правило может быть применено, если «условие» выполнено. Применение правила изменяет состояние рабочей памяти. Стратегия управления выбирает, какое именно правило из числа применимых следует использовать и прекращает вычисления, когда состояние рабочей памяти удовлетворяет целевому условию, либо ни одно из правил не может быть применено.

Основными достоинствами продукционных систем являются возможность использования механизма вывода в различных проблемных областях, естественность рассуждения, простота механизма логического вывода и объяснения хода вывода, а также создания, пополнения и модификации системы правил. К недостаткам следует отнести сложность оценки целостного образа знаний, описываемых системой правил, неясность взаимных отношений правил, отсутствие гибкости в логическом выводе, например, обработка нештатной ситуации, накладные расходы при проверке применимости каждого правила на каждом шаге вычисления.

Поэтому на практике продукционные системы дополняются системами, основанными на нейронных сетях, нечеткой логике, или комбинации этих подходов [3]. Это позволяет сочетать достоинства разных подходов. Например, нейронные сети применяются для решения трудноформализуемых задач.

3. Модуль задания правил

Предлагаемый модуль задания правил является частью продукционной системы. Базовая структура продукционной экспертной системы и ее основные элементы представлены на Рис. 1:

- База знаний - представление знаний в конкретной предметной области и управление ими;
- Модуль логического вывода - механизм, который на основании базы знаний осуществляет построение вывода;
- Модуль приобретения знаний, который служит для поддержки базы знаний и ее пополнения;
- Модуль задания правил обеспечивает получение знаний от эксперта и инженера по знаниям и передачу их в модуль приобретения знаний;
- Модуль объяснения способен не только давать заключение, но и представлять различные комментарии, прилагаемые к этому заключению, объяснять его мотивы;
- Пользовательский интерфейс необходим для взаимодействия системы с пользователем.

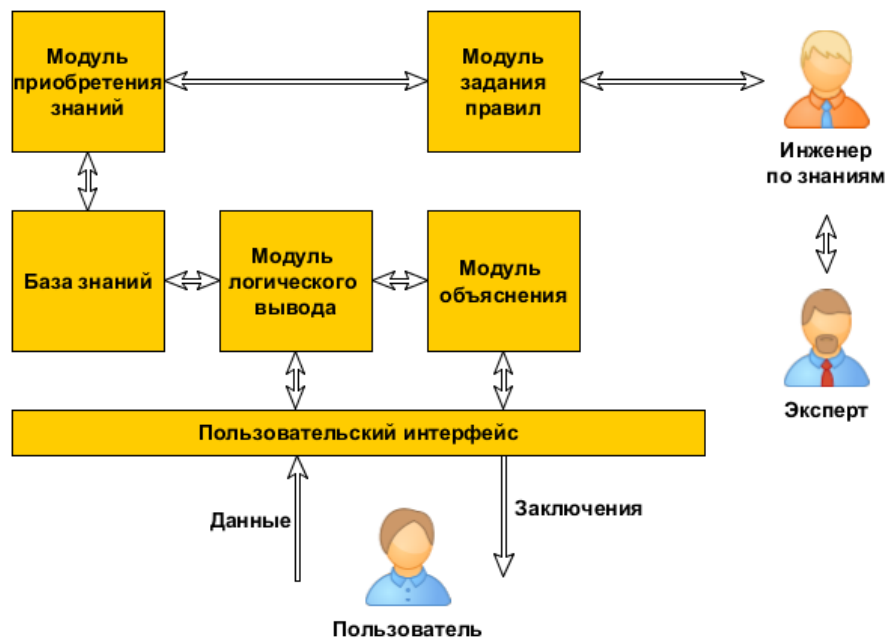


Рис. 1. – Основные модули продукционной системы

Предметная область рассматривается как набор объектов. Описание объекта, представленное в виде набора его свойств, будем называть фактом. Каждое свойство при этом является парой «название свойства»-«значение свойства».

Модуль задания правил для представления информации о предметной области использует файл специальной структуры. Этот файл имеет три части: задание шаблонов для описания фактов, задание фактов и задание правил.

Шаблон для описания фактов имеет название и состоит из перечисления именованных полей-свойств (слотов), которые могут быть задействованы при задании фактов этого типа. Описание каждого слота начинается с ключевого слова “slot”, далее указываются наименование слота, его тип и значение по умолчанию. Шаблоны могут наследоваться. Если один из шаблонов наследуется из другого, то в первый включаются все слоты из второго и дополнительные слоты, которые задаются явно при описании наследующего (расширяющего) шаблона. Для указания, что шаблон наследуется из другого, служит ключевое слово «extends» с именем наследуемого шаблона.

```

<ШаблонФакта> ::= <ИмяШаблона> [extends <ИмяШаблона>] <СписокСлотов>
<СписокСлотов> ::= <Слот> | <Слот><СписокСлотов>
<Слот> ::= slot <ИмяСлота> [type=<ТипСлота>] [default=<ЗначениеПоУмолчанию>]
<ТипСлота> ::= integer | float | string

```

Описания шаблонов используются для контроля правильности задания фактов.

Каждый факт имеет название, указание на используемый шаблон, и состоит из пар «название слота»-«значение слота для этого факта».

```

<Факт> ::= <ИмяФакта> <ИмяШаблонаФакта> <СписокЗначенийСлотов>
<СписокЗначенийСлотов> ::= <ЗначениеСлота> | <ЗначениеСлота> ;
<СписокЗначенийСлотов>
<ЗначениеСлота> ::= <ИмяСлота> = <ЗначениеДанногоСлота>

```

Правило состоит из названия, списка используемых шаблонов фактов и двух основных частей: задания условий применимости правила (секция “where”) и списка действий (секция “do”). Формат задания правил в таком виде распространен при разработке экспертных систем [4]. В настоящей работе используется следующий формат правила:

```
<Правило> ::= <ИмяПравила> <СписокШаблонов> where <УсловиеПрименимости>  
do <СписокДействий>
```

где <УсловиеПрименимости> – логическое выражение, а <СписокДействий> – изменения свойств фактов, добавление или их удаление. Область применения правила ограничивается теми фактами, которые описываются шаблонами из <СписокШаблонов>. Описание пользователя переводится в программу на языке программирования Python. В раздел <СписокДействий> также можно включать фрагменты программного кода с использованием синтаксиса Python, что обеспечивает дополнительные возможности по управлению логическим выводом. Выбор этого языка программирования определяется его гибкостью в описании данных и исполнении кода; в том числе, описан успешный опыт его применения для создания оболочек экспертных систем [5].

В качестве примера можно рассмотреть задачу выбора автомобиля с помощью соответствующей продукционной системы. Сначала рассмотрим заполнение базы знаний системы.

```
auto  
slot make  
slot model  
slot year type=integer  
slot color default=white
```

В этом примере задается шаблон для описания фактов с именем auto и слоты для указания производителя, модели автомобиля, года выпуска и цвета. По умолчанию цвет автомобиля – белый.

Далее задаются факты о конкретных автомобилях.

```
Fact1  
auto  
model=LeBaron; make=Chrysler; year=1997  
Fact2  
auto  
make=Toyota; model=Corolla; color=red; year=2003
```

После этого создается правило, например, выбора автомобиля не старше 2000 года выпуска.

```
Rule1  
auto  
where auto.year > 2000  
do print( str(auto.model) + " " + str(auto.year) )
```

Работа модуля задания правил состоит в следующем. После того, как пользователем заданы факты и правила, все части файла (задание шаблонов фактов, задание фактов на основе шаблонов, задание правил) проходят синтаксическую проверку и переводятся во внутреннее представление. На основе внутреннего

представления генерируется текст программы на языке Python. Факты переводятся в массив словарей, правила – в функции.

Результат работы модуля используется для организации вывода. Создается главная функция, определяющая последовательность вызова функций-правил. Работа системы состоит в последовательном вызове функций-правил. Работа заканчивается, когда достигается целевое состояние системы (успех), или становится невозможным применить ни одно из правил (неуспех).

Для проверки работы системы была использована модельная задача прорыва рубежа ракетной обороны (ЗРК) группой летательных аппаратов (ЛА). Задача ЗРК состоит в уничтожении всех ЛА, задача хотя бы одного ЛА – пересечь рубеж обороны. В данной задаче фактами являются совокупности свойств объектов (ЗРК и ЛА) – скорость перемещения, радиус поражения, вероятность поражения противника при нанесении удара. Правила задают поведение объектов – перемещение, нанесение ударов. Результатом работы является прогноз развития ситуации, кто из противников выигрывает и с какой вероятностью. Предложенный в работе подход был применен для быстрого задания и изменения входных параметров и правил вывода продукционной системы.

4. Заключение

На основе предложенного в работе подхода к заданию правил был реализован программный модуль задания правил. Разработанный модуль был применен в качестве составной части системы при решении модельной задачи прорыва рубежа ракетной обороны. Для повышения гибкости системы в дальнейшем планируется введение модульности правил, разделение правил вывода и данных о предметной области. Также представляется интересным введение коэффициентов уверенности фактов и правил, что позволит работать с нечеткими данными.

Литература

1. Осипов Г.С. Методы искусственного интеллекта – М.: Физ-матлит, 2011. – 296 с.
2. Boose J. H. A survey of knowledge acquisition techniques and tools //Knowledge acquisition. 1989. Т. 1. №. 1. С. 3-37.
3. Sahin S., Tolun M. R., Hassanpour R. Hybrid expert systems: A survey of current approaches and applications //Expert Systems with Applications. 2012. Т. 39. №. 4. С. 4609-4617.
4. Friedman-Hill E. J. et al. Jess, the java expert system shell //Distributed Computing Systems, Sandia National Laboratories, USA. 1997.
5. Frederiksen B. Applying expert system technology to code reuse with Pyke //Proceedings of the conference on the Python Conference (PyCon). 2008.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ ОПТИМИЗАЦИИ РАСПОЛОЖЕНИЯ НАРУЖНОЙ ТЕХНОЛОГИЧЕСКОЙ УСТАНОВКИ НА ОБЪЕКТЕ НЕФТЕПЕРЕРАБОТКИ С УЧЕТОМ МИНИМИЗАЦИИ ПОЖАРНЫХ РИСКОВ

Гудин Сергей Витальевич

Ghent University

Ghent, Belgium, E-mail: sergey.gudin@firerisks.ru

Шихалев Денис Владимирович

Академия Государственной противопожарной службы МЧС России

г. Москва, E-mail: evacsystem@gmail.com

Хабибулин Ренат Шамильевич

Академия Государственной противопожарной службы МЧС России

г. Москва, E-mail: kh-r@yandex.ru

Аннотация

В работе описывается проблема поиска оптимального расположения новой технологической установки на объектах нефтепереработки с использованием современных информационных систем. На основе генетических алгоритмов создана математическая модель. Представлена специальная целевая функция и основные генетические операции. На базе информационной системы «FireRisks» проведен анализ полученной модели, сделаны выводы о полученных результатах.

1. Введение

Объекты нефтепереработки в силу своей специфики имеют большую степень потенциальной пожарной опасности, так как на их территории находится значительное количество пожаровзрывоопасных веществ и происходят сложные технологические процессы. Пожары на таких объектах создают значительную угрозу жизни и здоровью персонала работающего на данных предприятиях, а так же людям, находящимся в непосредственной близости таких объектов. К производственным объектам нефтепереработки относятся производственные комплексы зданий, сооружений и наружных установок, расположенных на отдельной площадке предприятия и предназначенные для осуществления технологического процесса производства [1] связанного с переработкой или транспортировкой нефтепродуктов, таких как нефть, бензин, дизельное топливо, сжиженные углеводородные газы (СУГ) и др. Некоторые исследования показали, что материальный ущерб пожаров на нефть перерабатывающих мощностей может составлять до 50% материального ущерба от всех пожаров [2].

В настоящее время существует много систем количественной оценки расчетных величин пожарных рисков [2], которые могут определять опасные факторы пожара на территории производственных объектов и прилегающим к ним территориям, а так же определять расчетные величины пожарных рисков. В то же время эти системы не обладают специальными алгоритмами поддержки принятия решений по снижению расчетных величин пожарных рисков. В ситуации, когда на территорию действующего предприятия требуется установить новую технологическую установку, решение принимается на основе экспертного заключения. Для решения этой задачи с использованием информационных систем нужно рассчитать значения пожарных рисков

при всех возможных вариантах расположения новой технологической установки. Так как оценка пожарного риска достаточно трудоемкая процедура, охватывающая множество факторов, влияющих на её значение, перебор множества вариантов с разным расположением нового объекта, может занять значительное количество времени и ресурсов. В настоящее время все шире используются информационные системы созданные по модели SaaS (SoftasaService). В данной модели вычисления выполняются на удаленном сервере, для которого критически важно количество затрачиваемых вычислительных ресурсов и времени. Поэтому, для решения задачи оптимизации размещения новой технологической установки на объектах нефтепереработки была выбрана концепция генетических алгоритмов [3]. Этот метод быстрее классических методик и требует меньшего количества ресурсов. Его эффективность определена во многих исследованиях, например в работах [4-6]. Генетические алгоритмы используются для решения задач с большим количеством информации. Многие авторы подчеркивают их эффективность при решении комплексных задач [7, 8, 9].

Для решения задачи поиска оптимального расположения технологической установки на территории нефтеперерабатывающих объектов использовалась классическая модель генетических алгоритмов, предложенная в 1975 г. Джоном Холландом (JohnHolland) в Мичиганском университете [10], с дальнейшей её модификацией для повышения эффективности решения требуемой задачи.

2. Целевая функция

Одним из обязательных критериев использования генетических алгоритмов является целевая функция, которая отвечает за количественное представление эффективности найденных решений. Составленная функция отражает безопасность того или иного расположения новой технологической установки для людей работающих на рассматриваемом предприятии и проживающих на близлежащих территориях и состоит из суммы 2-х параметров.

1. Количество расчетных величин пожарных рисков на территории, которые не превышают приемлемые значения.
2. Параметр среднего отклонения недопустимых расчетных величин пожарных рисков на территории нефтеперерабатывающего объекта и прилегающей к нему жилой зоне от приемлемых значений.

Количество приемлемых величин пожарных рисков на территории нефтеперерабатывающих объектов находится по следующие формуле:

$$K = \sum_{j=1}^J (\alpha(R_j)) + \sum_{m=1}^M (\beta(I_m)) + \gamma(S), \quad (1)$$

Где

$$\alpha(x) = \begin{cases} 1, & x \leq R_a; \\ 0, & x > R_a; \end{cases} \quad (2)$$

$$\beta(x) = \begin{cases} 1, & x \leq I_a; \\ 0, & x > I_a; \end{cases} \quad (3)$$

$$\gamma(x) = \begin{cases} 1, & x \leq S_a; \\ 0, & x > S_a; \end{cases} \quad (4)$$

R_j – итоговое значение индивидуального пожарного риска для работников предприятия;

J – количество работников на предприятии;
 I_m – итоговое значение индивидуального пожарного риска для людей, находящихся в жилой зоне, общественно-деловой зоне или зоне рекреационного назначения вблизи объекта;
 M – количество объектов с людьми в селитебной зоне прилегающей к предприятию;
 S – итоговое значение величины социального пожарного риска для людей, находящихся в жилой зоне, общественно-деловой зоне или зоне рекреационного назначения вблизи объекта;
 R_a – допустимое значение величины индивидуального пожарного риска для работников предприятия (равняется 10^{-6} год $^{-1}$ по российскому законодательству [11]);
 I_a – допустимое значение величины индивидуального пожарного риска для людей, находящихся в жилой зоне, общественно-деловой зоне или зоне рекреационного назначения вблизи объекта (равняется 10^{-8} год $^{-1}$ по российскому законодательству [11]);
 S_a – допустимое значение величины социального пожарного риска для людей, находящихся в жилой зоне, общественно-деловой зоне или зоне рекреационного назначения вблизи объекта (равняется 10^{-7} год $^{-1}$ по российскому законодательству [11]);
 α – критерий приемлемости значения величины индивидуального пожарного риска для работников предприятия;
 β – критерий приемлемости значения величины индивидуального риска для людей, находящихся в жилой зоне, общественно-деловой зоне или зоне рекреационного назначения вблизи объекта;
 γ – критерий приемлемости значения величины социального пожарного риска для людей, находящихся в жилой зоне, общественно-деловой зоне или зоне рекреационного назначения вблизи объекта;
 K – количество величин пожарных рисков являющихся приемлемыми в рассматриваемом случае.

Следующий параметр отражает среднее отклонении недопустимых расчетных величин пожарных рисков на нефтеперерабатывающем объекте и прилегающей к нему селитебной зоне от приемлемых значений. Данный параметр принимает значение от 0 до 1го и используется в случае когда не все значения пожарных рисков являются приемлемыми и служит для поиска точек территории где значения рисков будут наиболее сильно приближены к допустимым значениям:

$$Q = \frac{A + B + C}{Z + Y + \gamma(S)} \quad (5)$$

Где

$$A = \frac{\sum_{z=1}^Z R_a / R_z}{Z} \quad (6)$$

$$B = \frac{\sum_{y=1}^Y I_a / I_y}{M} \quad (7)$$

$$C = \begin{cases} \frac{S_a}{S}, & S > S_a; \\ 0, & S \leq S_a \end{cases} \quad (8)$$

A – безразмерный параметр среднего отклонения недопустимых величин индивидуальных рисков на территории предприятия (меньших R_a) от допустимого значения (R_a);

R_n – Значения величин недопустимых индивидуальных рисков на территории предприятия;

Z – Количество недопустимых значений величин индивидуальных рисков на территории предприятия;

B – безразмерный параметр среднего отклонения недопустимых величин индивидуальных рисков в селитебной зоне (меньших I_a) от допустимого значения (I_a);

I_y – Значения недопустимых расчетных величин индивидуальных рисков в селитебной зоне;

Y – Количество недопустимых расчетных величин индивидуальных рисков в селитебной зоне;

C – Безразмерный параметр отклонения величины социального риска в селитебной зоне от допустимого значения.

Общий вид целевой функции можно представить в следующем виде:

$$f(x, y) = K + Q \rightarrow \max, \quad (9)$$

где x, y – координаты рассматриваемой точки территории.

3. Начальная популяция

Для практической реализации рассматриваемого подхода, планируется использовать интернет-картографические системы, которые позволяют загружать свои карты (схемы, подложки), устанавливать масштаб, наносить дополнительные слои моделирования зон распространения рисков.

Чтобы работать с этой системой необходимо сначала вычислить координаты краев подложки, затем принять нижний левый угол за точку отсчета (0,0) и преобразовать остальные координаты в метрическое измерение. Таким образом правый верхний угол будет содержать максимальные значения возможных координат (x_{max}, y_{max}).

Для реализации генетического алгоритма необходимо закодировать оптимизируемые параметры в двоичные строки, таким образом, чтобы при декодировании число не могло получиться больше заданного максимума (размера территории объекта). Длина строки зависит от размера максимума.

Например, пусть переменная x имеет интервал изменения $[0, X_{max}]$. Требуемое число битов m_x находится по формуле:

$$X_{max} \leq 2^{m_x}, \quad (11)$$

$$m_x = \lceil \log_2(x_{max}) \rceil. \quad (12)$$

Обратное преобразование строки битов в действительное значение переменной выполняется по следующей формуле:

$$x_j = \lfloor l_j * X_{max} / 2^{m_x} \rfloor, \quad (13)$$

где l_j представляет собой десятичное значение (ген) закодированной в бинарной системе координаты.

Для координаты “ y ” проводятся операции по аналогии с координатой “ x ”.

Исходная популяция (популяции) генерируется случайно. И её генотип v имеет вид:

$$v = [(2^{m_x})_2 (2^{m_y})_2]. \quad (14)$$

4. Функция соответствия

Функция соответствия оценивает хромосомы по степени их приспособленности к выполнению критерия оптимизации.

Оценка функции соответствия хромосомы выполняется в три шага:

1. Преобразовать генотип хромосомы в фенотип. В данной задаче это означает поиск мероприятий в базе данных по идентификаторам в хромосоме $n^k=(M^k)$, $k=1,2,\dots, pop_size$, где pop_size - число мероприятий в исходной популяции.

2. Вычислить целевую функцию.

3. Преобразовать целевую функцию в значение функции соответствия. Для решаемой задачи оптимизации функция соответствия эквивалентна целевой функции.

$$eval(v_k) = f(n^k), k = 1, 2, \dots, pop_size. \quad (15)$$

5. Отбор

Для отбора был выбран подход, называемый колесо рулетки [12] (от англ. roulette wheel). Согласно этому подходу отбор осуществляется на основе некоторой функции распределения, которая строится пропорционально вычисленным функциям соответствия сгенерированных вариантов-хромосом. Колесо рулетки может быть сконструировано следующим образом:

1. Вычисляем значение функции соответствия $eval(v_k)$ для каждой хромосомы v_k по предыдущей формуле.
2. Вычисляем общую функцию соответствия популяции:

$$F = \sum_{k=1}^{pop_size} eval(v_k) - j = 1, pop_size \{eval(v_j)\}. \quad (16)$$

3. Вычисляем вероятность отбора P_k для каждой хромосомы v_k :

$$P_k = \frac{eval(v_k) - j = 1, pop_size \{eval(v_j)\}}{F}, k = 1, 2, \dots, pop_size. \quad (17)$$

4. Вычисляем совокупную вероятность q_k для каждой хромосомы v_k :

$$q_k = \sum_{j=1}^k P_j, k = 1, 2, \dots, pop_size. \quad (18)$$

Процесс отбора начинается с вращения колеса pop_size раз; при этом каждый раз выбирается одна хромосома по следующему алгоритму:

1. Генерируем случайное число r из интервала $[0,1]$;
2. Если $r \leq q_1$, то выбираем первую хромосому v_1 ; иначе выбираем k -ую хромосому ($2 \leq k \leq pop_size$) такую, что $q_{k-1} \leq r \leq q_k$.

6. Скрещивание и мутация

Для скрещивания хромосом используется популярный на практике метод с одной точкой обмена [13, 14]. В соответствие с этим методом случайно выбирается одна точка обмена, относительно которой меняются местами части хромосом-

родителей. Для этого генерируется целое число в промежутке $[1, (m_x + m_y)]$, которое будет точкой обмена генами.

Мутация состоит в изменении одного или большего числа генов с вероятностью равной коэффициенту мутации. Так как планируется использовать метод “инцеста”, этот коэффициент будет увеличиваться пропорционально родству родительских особей, чтобы избежать попадания алгоритма в локальный оптимум. Так как мы имеем дело с бинарными строками, то мутация заключается в инверсии соответствующего бита.

7. Анализ

Для анализа полученного подхода модель была реализована в виде специального модуля информационной системы по управлению пожарными рисками “*FireRisks*” (www.firerisks.ru) [15]. В качестве производственного объекта, на котором проводится пробное расчётное тестирование, рассматривается проект условной газораспределительной станции (ГРС).

В качестве производственного объекта, на котором проводится пробное расчётное тестирование, рассматривается проект условной газораспределительной станции (ГРС).

Основными технологическими операциями, осуществляемыми на ГРС являются, приём, хранение, перекачивание и отгрузка сжиженных углеводородных газов (СУГ). В состав ГРС входят следующие основные наружные технологические установки, здания и сооружения (рис. 1):

- резервуарный парк СУГ, состоящий из резервуаров горизонтальных стальных (РГС) вместимостью от 50 до 100 м³;

- сепаратор;

- автомобильная и железнодорожная сливо-наливная эстакада СУГ;

- административно-бытовой комплекс (АБК) и операторные.

Объекты на территории:

Зона А – зона работы операторов на железнодорожной эстакаде, 2 человека;

Зона Б – зона работы операторов у резервуарной группы (РГС 50), 2 человека;

Зона В – зона работы операторов у резервуарной группы (РГС 100), 2 человека;

Зона Г – зона работы операторов у резервуарной группы (РГС 100), 2 человека;

Зона Д – зона работы операторов у сепаратора, 1 человек;

Зона Е – зона работы операторов на автогазозаправочной станции (АГЗС), 2 человека;

Зона Ж – административно-бытовой комплекс (АБК), 8 человек;

Прилегающие объекты:

Зона З – торговый комплекс, 1000 человек;

Зона И – микрорайон “Нефтяник”, 300 человек.

Для оценки эффективности использования генетических алгоритмов при решении задачи поиска оптимального расположения новой технологической установки на территории нефтеперерабатывающего объекта рассмотрена ситуация, заключающаяся в необходимости расположения одного дополнительного резервуара РГС 50 на территории предприятия.

Эксперимент состоял из нескольких этапов. На первом этапе произведен поиск оптимального расположения новой технологической установки на территории нефтеперерабатывающего объекта при помощи линейного алгоритма поиска (рассмотрение каждого метра территории как возможного оптимума) (рис. 1). Красным цветом зоны содержащие оптимальные или близкие к оптимуму значению, глобальный оптимум помечен синим маркером “Optimum”. Несмотря на точность линейного поиска, этот алгоритм требует большого количества времени и компьютерных ресурсов. Общее

время выполнения моделирования составило *1 час 16 минут 36 секунд*, при следующих характеристиках компьютера:

- материнская плата: ASUS X99-A (RTL) LGA2011-3;
- процессор CPU Intel Core i7-5820K 3.3 GHz / 6 core;
- оперативная память: DDR4 PC4-17000 (4x8 Gb).

Общее количество рассматриваемых точек составило *71700*. Необходимый объем оперативной памяти для выполнения эксперимента составил *28,735 мегабайт*. Исходя из этого, можно сделать вывод, что в настоящее время для выполнения линейных расчетов по поиску оптимального расположения нового технологического объекта с точностью 1м является сложной задачей в виду большого потребления компьютерных ресурсов и времени.

Следующим этапом эксперимента являлся поиск оптимального расположения новой технологической установки при помощи предложенного генетического алгоритма, критерием остановки было определено ограниченное количество популяций. Так как количество рассматриваемых точек должно зависеть от размера территории, количеством популяций являлась сумма ширины и длины территории. Размер популяции определялся суммой битов зашифрованных строк умноженных на 2.

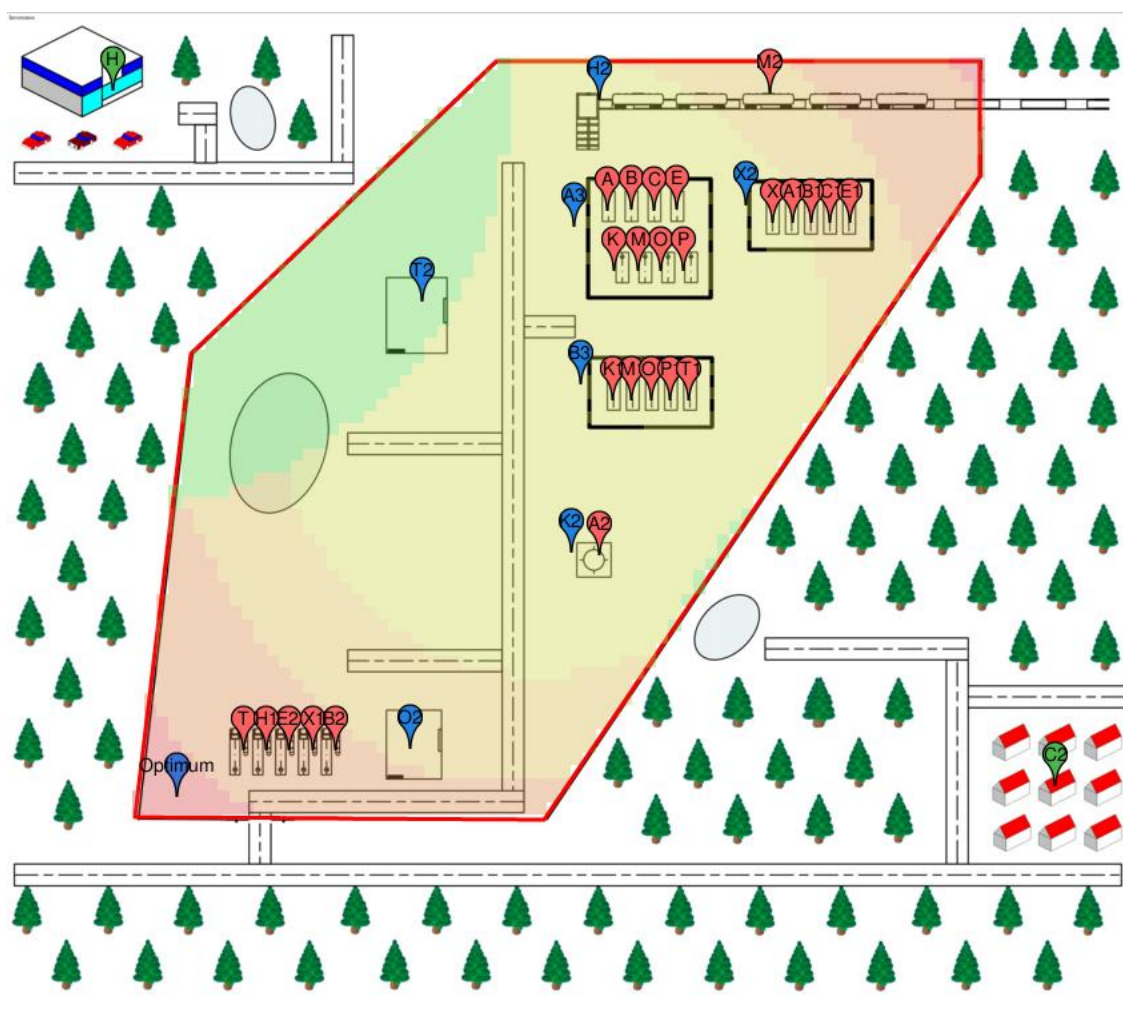


Рис. 1. – Линейный поиск оптимального расположения нового РГС 50

Эксперимент № 1

Количество мутируемых особей: 20%; количество особей подвергаемых скрещиванию: 50%. После серии экспериментов, были выявлены случаи когда

алгоритм “застревал” в локальных зонах и рассматривал ложные зоны как содержащие оптимум.

Эксперимент № 2

Для охвата большего количества зон территории процент мутирующих особей был увеличен до 50%, а количество скрещивающихся особей до 90%. В результате этого, в каждом из 10 экспериментов все зоны содержащие локальные оптимумы были рассмотрены. Одним недостатком являлось излишне большое количество операций производимых в зонах локальных оптимумов и малое количество операций других, что может в некоторых случаях привести к определению локального оптимума как глобального.

Эксперимент № 3

Для решения вышеуказанной проблемы был использован метод инцеста [17] (рис. 2). При использовании этого метода проблема неравномерного количества операций в зонах содержащих локальные оптимумы была решена.

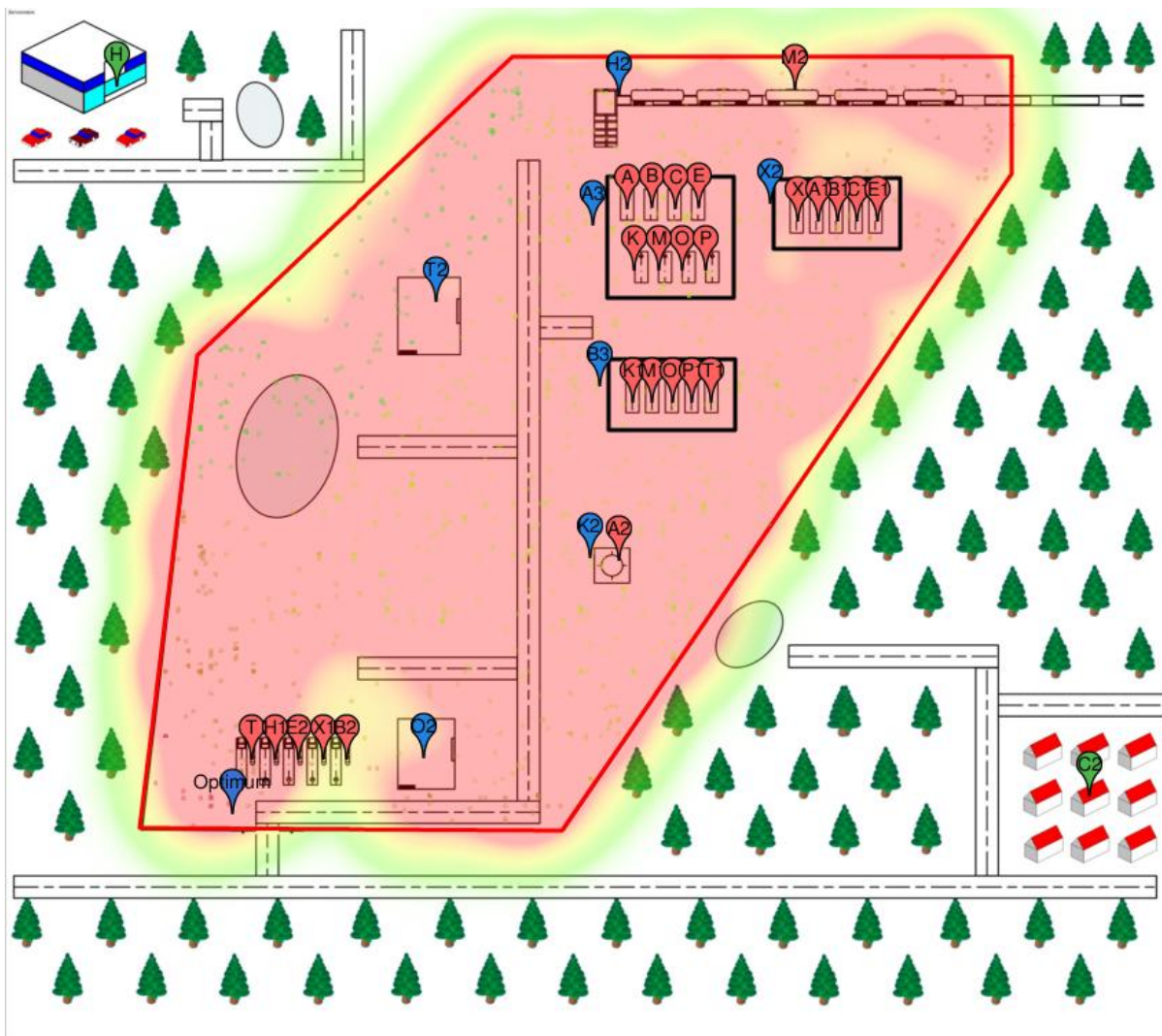


Рис. 2. – Использование метода инцеста для поиска глобального оптимума

Эксперимент № 4.

Так как при поиске локального оптимума при помощи линейного алгоритма погрешность составляла 1м, а при использовании генетических алгоритмов эта погрешность составляет 5м, то сравнивать их скорость работы этих алгоритмов нельзя.

Для возможности сравнения этих алгоритмов был проведен поиск оптимального расположения нового технологического объекта при помощи линейного алгоритма с погрешностью 5 м.

Результаты всех экспериментов представлены в таблице 1, а так же визуально изображены на рис.3 .

Таблица 1

Результаты поиска оптимального расположения нового технологического объекта на территории нефтеперерабатывающего объекта с использованием различных методов

№	Использование генетических алгоритмов	Количество мутуируемых особей, %	Количество особей подвергаемых скрещиванию, %	Количество рассмотренных точек, ед.	Погрешность, м	Использование метода инцеста	Время, мин.
1	да	20	50	1083	19	нет	0:43
2	да	50	90	874	5	нет	0:41
3	да	50	90	852	5	да	0:38
4	нет	-	-	2868	5	нет	3:02

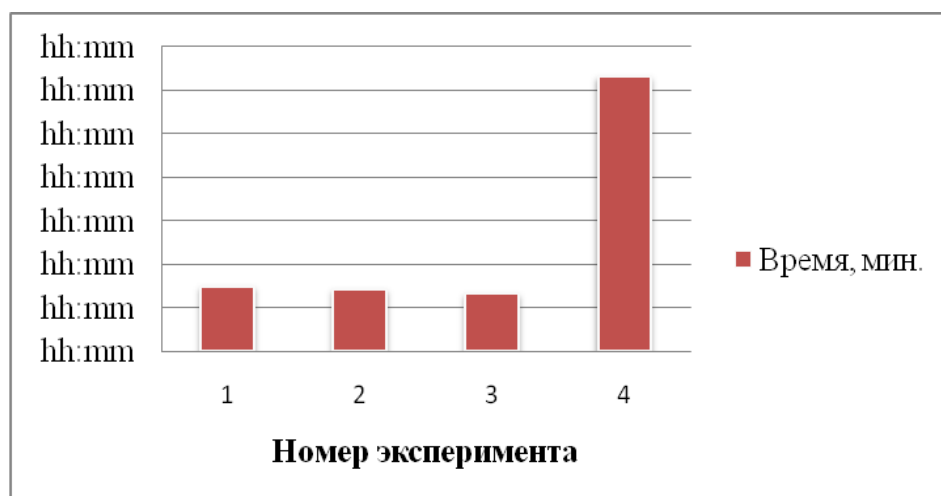


Рис. 3. – Сравнение количества требуемого времени при использовании разных методов поиска оптимального расположения новой технологической установки

В результате сравнения полученной модели с линейным перебором всех возможных вариантов расположения новой технологической установки, сделаны следующие выводы:

1. Количество рассматриваемых вариантов расположения технологической установки *уменьшилось в 3,4 раза*;

2. Требуемое время *сократилось в 4,8 раза*.

В свою очередь линейный алгоритм обладает одним преимуществом, заключающемся в возможности визуализировать возможные зоны установки новой технологической установки.

8. Заключение

Таким образом, можно сделать вывод что линейный и генетический алгоритмы эффективны при решении разных задач. В задаче, когда требуется найти зону, содержащую глобальный оптимум генетический алгоритм предпочтительней линейного за счет потребления меньшего количества времени и ресурсов. Для поиска массива локальных оптимумов, он так же эффективен.

В задаче, когда необходимо визуализировать возможные зоны установки новой технологической установки необходимо использовать линейный алгоритм, так как генетический алгоритм в виду особенности выборочного анализа территории не может визуализировать эти зоны.

Направлением дальнейшей работы является модификация созданной модели с целью улучшения скорости и качества её работы, а так же создание других моделей направленных на поддержку принятия решений при управлении пожарными рисками на объектах нефтепереработки.

Литература

1. Еремин Н. А., Кондратюк А. Т., Еремин А. Н. Ресурсная база нефти и газа арктического шельфа России // Георесурсы, геоэнергетика, геополитика. – 2010. – С. 2009-1.
2. Гудин С. В., Хабибулин Р. Ш., Рубцов Д. Н. Проблемы управления пожарными рисками на территории объектов нефтепереработки с использованием современных программных продуктов // Пожаровзрывобезопасность. — 2015. — Т. 24, № 12. — С. 40-45. DOI: 10.18322/PVB.2015.24.12.40-45.
3. Бакаев М.А., Авдеенко Т.В. X Всероссийская заочная конференция по теоретическим основам проектирования и разработки распределенных информационных систем (ПРИС-2012) // Индексирование и сопоставление сущностей предметной области в интеллектуальной системе на основе онтологического подхода. 2012. С. 6.
4. Panov N.V., Shary S.P. Interval evolutionary algorithm search the global optimum // News of Altai State University, No. 12.
5. Schaefer I.A. - Youth and Science: Proceedings of the VIII All Russian scientific and technical conference of students, graduate students and young scientists dedicated // Investigation of the efficiency of genetic algorithm constrained optimization. 2012.
6. Sergienko R.B. Efficacy coevolutionary genetic algorithm constrained optimization // Herald Siberian State Aerospace University. Academician M.F. Reshetnev, No. 3, 2009.
7. Martorell S., Villanueva J.F., Carlos S., Nebot Y., Sánchez A., Pitarch J.L., Serradell V. RAMS+ C informed decision-making with application to multi-objective optimization of technical specifications and maintenance using genetic algorithms // Reliability Engineering & System Safety, Vol. 87, No. 1, 2005. pp. 65-75.
8. Ramirez A.J., Knoester D.B., Cheng B.H., McKinley P.K. Proceedings of the 6th international conference on Autonomic computing // Applying genetic algorithms to decision making in autonomic computing systems. 2009.
9. Caputo A.C., Pelagagge P.M., Palumbo M. Economic optimization of industrial safety measures using genetic algorithms // Journal of Loss Prevention in the Process Industries, Vol. 24, No. 5. pp. 541-551.
10. Holland J.N. Adaptation in Natural and Artificial Systems // Univ. Michigan Press, 1975.
11. Приказ МЧС РФ от 10.07.2012 г. № 404 «Об утверждении методики определения расчетных величин пожарного риска на производственных объектах».

12. Шефер И.А. Молодёжь и наука: Сборник материалов VIII Всероссийской научно-технической конференции студентов, аспирантов и молодых учёных // Исследование эффективности генетического алгоритма условной оптимизации. 2012.
13. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ.
14. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. М.: ФИЗМАТЛИТ, 2003. 264-275 pp.
15. Гудин С.В., Хабибулин Р.Ш. XV Всероссийская конференция молодых ученых по математическому моделированию и информационным технологиям // Алгоритм управления пожарной безопасностью на нефтегазовых объектах с использованием информационной системы FireRisks. Тюмень. 2014. С. 81.
16. Desideri J.A., Peigin S., Timchenko S. Application of genetic algorithm to space vehicle reentry trajectory optimization problem // Inria, No. 3843, 1999

СЕТИ ТРАНСФЕРА ТЕХНОЛОГИЙ КАК ЭЛЕМЕНТ ОЦЕНКИ УРОВНЯ ТЕХНОЛОГИЗАЦИИ РЕЗУЛЬТАТОВ ПРИКЛАДНЫХ ИССЛЕДОВАНИЙ¹

Каменская М.А.

Институт системного анализа Федерального исследовательского центра «Информатика и управление» Российской академии наук
г. Москва, mak@isa.ru

Тихомиров И.А.

Институт системного анализа Федерального исследовательского центра «Информатика и управление» Российской академии наук
г. Москва, tih@isa.ru

Ядринцев В.В.

Общество с ограниченной ответственностью «Технологии системного анализа»
г. Москва, vvyadrincev@gmail.com

Рутковская И.Б.

Фонд информационного обеспечения науки
г. Москва, t5817951@yandex.ru

Аннотация

В статье рассмотрена задача оценки уровня технологизации результатов прикладных исследований, проанализированы известные российские и зарубежные сети трансфера технологий как потенциально возможный источник информации для проведения научно-технологической экспертизы. Предложен ряд критериев, значения которых вычисляются на основе тематической и содержательной близости технологических запросов и технологических предложений, хранящихся в сетях трансфера технологий. Предложенные критерии могут быть использованы как элемент оценки уровня технологизации научных результатов.

¹ Работа выполнена при поддержке Минобрнауки России (контракт № 2015-12), и РФФИ (проект №14-29-05075 «офи_м»)

1. Введение

Под уровнем технологизации (или технологизируемости) в широком смысле принято понимать степень готовности результата исследований к внедрению, использованию или созданию некоторого продукта или промышленной технологии. В США для определения уровня готовности технологий принято использовать шкалу Technology Readiness Levels (TRL), разработанную NASA [1]. Эта шкала описывает 9 уровней готовности технологии:

1. Рассмотрены и опубликованы основные принципы технологии.
2. Сформулированы основные концепции технологии и возможности ее прикладного использования.
3. Выполнены аналитические и экспериментальные исследования, доказывающие правильность концепций, предложенных на Уровне 2.
4. Проведена проверка компонента или макета в лабораторных условиях.
5. Проведена проверка компонента или макета в условиях, имитирующих реальные.
6. Демонстрация прототипа системы или подсистемы в условиях, имитирующих реальные.
7. Демонстрация прототипа системы в реальных условиях.
8. Завершена разработка системы, проведены испытания в реальных условиях.
9. Проведена интеграция разработки в существующую систему.

Технологии с первого по шестой уровень соответствуют этапу научно-исследовательских работ, а начиная с седьмого уровня – этапу опытно-конструкторских работ.

На сегодняшний день шкала TRL взята за основу оценки уровня технологизации многими ведомствами, например, Министерством обороны США, Федеральным управлением гражданской авиации США, Европейским космическим агентством, различными организациями Европейского союза, и в различных областях: промышленность, программное обеспечение, биомедицинские технологии, авиация, энергетика и др. Причем, практически во всех сферах перечень возможных уровней разработок остается неизменным, меняется только содержательный аспект каждого уровня. Использование классификации TRL упрощает разработчикам и заказчикам контроль за выполнением исследовательских работ и помогает выбрать наиболее готовые к внедрению технологии.

Следует отметить, что достоверная и обоснованная оценка уровня технологизации результатов научных исследований может быть полезна научным фондам, задачей которых является проведение отбора лучших инициативных проектов и дальнейшая их финансовая поддержка. Технологизация может рассматриваться как один из критериев конкурсного отбора заявок на получение финансовой поддержки.

Традиционный путь получения объективной оценки результата исследований – это проведение экспертизы. На сегодняшний день в качестве дополнительных источников информации при проведении научно-технологической экспертизы используются электронные научные базы данных. В современных реалиях в связи с лавинообразным ростом количества информации у экспертов, оценивающих перспективность ожидаемых результатов прикладных научных исследований, возрастает потребность в информационных системах, предоставляющих функции поиска и систематизации информации, представленной в крупных коллекциях научно-технических документов, а также функции автоматического анализа

документов этих коллекций. В данном исследовании предлагается использовать сети трансфера технологий в качестве дополнительного источника информации для проведения научно-технологической экспертизы прикладных исследований, а именно для оценки уровня технологизации результатов научно-технической деятельности (НТД).

Сети трансфера технологий начиная с середины 1990-х годов стали одним из основных институтов коммерциализации результатов научной деятельности, позволяющих эффективно систематизировать и распространять информацию о технологиях и осуществлять поиск партнеров в области создания и практической реализации новых технологических разработок. Таким образом, сети трансфера технологий являются источником большого количества информации, с помощью которой можно оценивать востребованность и перспективность результатов прикладных научных исследований, а также уровень их технологизации. Основные функциональные возможности крупнейших существующих сетей трансфера технологий будут рассмотрены во втором разделе статьи. В третьем разделе будут предложены критерии оценки уровня технологизации и востребованности результатов прикладных научных исследований, а также технологии анализа информации, полученной из сетей трансфера технологий.

2. Функциональные возможности сетей трансфера технологий

В общем случае сеть трансфера технологий развивается как структура распределенного типа, где каждый член сети взаимодействует со своими партнерами и клиентами. Клиенты сети – это поставщики и потребители новых технологий. Клиентами обычно становятся компании малого, среднего и крупного бизнеса, научно-исследовательские институты, университеты, частные лица, нацеленные на продвижение новых технологий и поиск технологических партнеров. Партнерами сети являются различные организации, поддерживающие развитие сети, например, путем участия в совместных проектах. Разделение на поставщиков и клиентов довольно условно, поскольку одна и та же компания или частное лицо может одновременно выступать как в роли поставщика, так и в роли клиента.

Основной задачей членов сети трансфера технологий является поиск партнеров и установление сотрудничества между заинтересованными сторонами (продавцом и покупателем технологии) с целью дальнейшего развития и/или коммерческого использования новых технологий. При этом члены сети оперируют двумя ключевыми понятиями: технологическое предложение и технологический запрос. Оба понятия, как правило, представляют собой определенную структурированную анкету, которая предоставляет потенциальным партнерам информацию, достаточную для быстрого ознакомления с предлагаемой технологией (технологическим предложением) или для понимания технологической потребности компании (технологическим запросом), включая коммерческие аспекты, степень инновационности, главные преимущества, зарегистрированные права интеллектуальной собственности и другие характеристики желаемого партнерства.

Поиск партнеров и установление сотрудничества достигается путем сопоставления имеющихся технологических предложений с выявленными технологическими запросами. При этом все члены трансферной сети должны использовать единую методологию работы по трансферу технологий. Для осуществления поиска партнеров и продвижения технологической информации в сети используются такие инструменты, как:

- единая база технологических запросов и предложений;

- технологический аудит, представляющий собой проверку предложений и запросов на техническую состоятельность и готовность к трансферу;
- веб-сайт как информационный ресурс, на котором представлена единая база технологических запросов и предложений;
- периодическая рассылка участникам информации о деятельности сети и её новых профилях;
- сетевое взаимодействие с членами и партнерами сети [2].

Важной возможностью сетей трансфера технологий является сбор статистики обращений клиентов сети к профилю, с помощью которой можно анализировать динамику изменения спроса потребителей технологий на предложения, а также отслеживать инновационный потенциал технологий в зависимости от времени.

а. Европейская сеть трансфера технологий

На сегодняшний день в мире существует множество сетей трансфера технологий схожих по своим целям, но не всегда схожих по основным функциональным возможностям. Одной из первых (в 1995 г.) по инициативе Европейской Комиссии была создана Европейская сеть центров передачи инноваций IRC (Innovation Relay-Centers) с целью содействия малым и средним европейским компаниям в сфере трансфера технологий [3]. Сеть IRC действовала до 2008 г., после чего ее функции перешли к Европейской сети поддержки предпринимательства (Enterprise Europe Network, EEN) [4]. Enterprise Europe Network – европейская сеть трансфера технологий, объединяющая около 600 организаций поддержки предпринимательства, таких как палаты торговли и промышленности, технологические центры, научно-исследовательские институты, университеты и агентства по развитию.

Информационная база данных EEN – одна из крупнейших в Европе в области передовых технологий, содержит более 23 тыс. профилей технологических запросов и предложений из более чем 60 стран мира. Доступ к базе EEN и работа с ней осуществляются при помощи информационно-технологической платформы Merlin [5]. Merlin является инструментом автоматизации наиболее важных бизнес-процессов работы центров EEN, включая размещение и поиск профилей в единой базе данных, сопровождение выражений интереса, организацию и участие в различных мероприятиях, организуемых в рамках сети, работу с запросами клиентов и др.

Сервис предоставляет функцию поиска профилей по запросу и набору критериев: тип профиля, страна, в которой составлен профиль, и страна, в которой клиент, посылающий запрос, ищет сотрудничества. Профили подразделяются на 5 типов:

- бизнес-предложение – профили, предлагающие продукты или сервисы;
- бизнес-запрос – профили, запрашивающие продукты или сервисы;
- технологическое предложение – профили, предлагающие знания или опыт;
- технологический запрос – профили, запрашивающие знания или опыт;
- запрос исследования развития – профили, в которых ведется поиск исследований развития совместных проектов.

Поиск осуществляется в двух режимах: в выдаче результатов поиска могут содержаться профили, в которых встречаются обязательно все ключевые слова из запроса или только некоторые слова из запроса. Также результаты выдачи могут выводиться в сжатом (компактном) или в подробном виде. При выборе компактного вида для каждого профиля на страницу выводится заголовок, страна, тип и дата создания профиля. В подробной информации к вышеназванным полям добавляется краткая аннотация. Результаты выдачи можно сортировать по дате последнего изменения профиля.

Каждый профиль можно открыть на отдельной странице для более подробного изучения. В деталях профиля содержатся:

- заголовок профиля;
- аннотация;
- описание предлагаемых продуктов/сервисов/технологий;
- преимущества инновации;
- стадия развития инновации и комментарии относительно нее (например, «готово к серийному производству; небольшая партия образцов уже успешно выпущена»);
- права на интеллектуальную собственность;
- ключевые слова;
- описание искомого партнера;
- информация об авторах профиля.

Ключевые слова делятся на 3 типа: технологические, рыночные, производственные. Ознакомиться с подробной классификацией ключевых слов и их кодов можно в приложениях руководства пользователя платформы Merlin [5].

Искомый партнер в свою очередь описывается по пяти параметрам:

- тип требуемого сотрудничества (например, договор о предоставлении услуг и лицензионное соглашение);
- тип искомого партнера (промышленная организация/научно-исследовательские институты и др.);
- область деятельности партнера;
- задачи, стоящие перед партнером;
- предпочитаемые страны (не обязательный параметр).

Больше возможностей для работы с профилями имеют зарегистрированные клиенты сети, т.е. клиенты, создавшие личную учетную запись и заполнившие контактную информацию. Например, есть возможность экспорта интересующего профиля в форматы pdf, html, docx, xlsx и xml для дальнейшей локальной работы с профилем на компьютере или передачи профиля другим заинтересованным лицам.

Также зарегистрированные пользователи имеют преимущество в процессе обращения (или выражения интереса) к профилю: достаточно авторизоваться в системе перед началом поиска подходящих профилей, а в процессе обращения к профилю нажать нужную кнопку и ответить на несколько вопросов касательно интересующего профиля. Незарегистрированному пользователю придется для каждого обращения заполнять поля с контактной информацией. В личном кабинете можно просмотреть историю всех обращений к профилям.

Еще одна полезная функция, предоставляемая зарегистрированным пользователям – создание запросов, основанных на наборе критериев, значение которых пользователь задает согласно своим предпочтениям к технологическим профилям. Эти запросы не аналогичны обычному поисковому запросу, так как они сохраняются и редактируются в личном кабинете пользователя сети, и на основе этих сохраненных запросов пользователь с определенной периодичностью получает новостную рассылку с новыми профилями или предстоящими событиями сети, которые, возможно, будут представлять для него интерес.

в. Российская сеть трансфера технологий

Опыт в области трансфера технологий, наработанный сетью IRC, успешно заимствовала крупнейшая Российская сеть трансфера технологий (Russian Technology Transfer Network, RTTN), созданная в 2002 г. и объединяющая более 50 инновационных центров из 40 регионов России и стран СНГ, специализирующихся в сфере трансфера

технологий [6]. Этот проект был инициирован Региональным инновационным технологическим центром (РИТЦ) наукограда Обнинска в партнерстве с Инновационным Центром наукограда Кольцово (ИЦК) в рамках реализации проекта TACIS FINRUS 9804 «Инновационные центры и наукограды Российской Федерации». В ходе проекта инновационные центры наукоградов установили партнерские отношения с центрами передачи инноваций Европейской сети IRC. С 2008 г. RTTN входит в состав сети поддержки бизнеса и инноваций EEN [4].

RTTN имеет четко выраженную ориентацию на внутринациональный трансфер технологий. Транснациональный трансфер технологий реализуется через участие RTTN в специальных международных проектах. Членство в сети открыто для всех организаций соответствующего профиля, однако заявители проходят обязательную сертификацию – процедуру проверки соответствия ее деятельности ряду критериев.

Информационно-технологическая платформа (ИТП) RTTN, которая развивалась в соответствии с потребностями членов сети, их клиентов и необходимостью её координации/управления, является инструментом, позволяющим автоматизировать работу региональных центров – членов сети, осуществлять мониторинг их работы и информационный обмен между ними. ИТП является единым информационным пространством участников RTTN [7] и включает в себя:

- информационные базы данных,
- интерфейс авторизованного доступа к данным через Интернет, интерфейсы широкого (свободного) доступа – сайты RTTN, ее международных и региональных сегментов,
- регламенты и процедуры работы центров с информационной системой, программное и методическое обеспечение.

Каждый сертифицированный центр-участник RTTN получает от координирующей организации сети авторизованный доступ к администраторской части, где ему предоставляется рабочее место технологического брокера и возможность размещать от своего имени профили инновационных технологий, за качество и состоятельность которых он несет полную ответственность. Технологические профили, размещаемые в системе RTTN, предоставляют информацию по такому же шаблону, как в описанной выше сети EEN. Подробнее структура технологического профиля сети RTTN описана в руководстве по подготовке технологических предложений и запросов для сети RTTN [7].

Администраторская часть сайта RTTN реализована на русском и английском языках и включает в себя базы данных, содержащие следующую информацию:

- о клиентах центров RTTN, их научно-технических компетенциях, технологических запросах и предложениях, инновационной продукции;
- новостную информацию и информацию о событиях, в которых могут принять участие члены RTTN и их клиенты;
- информацию о российских и зарубежных программах и конкурсах поддержки научно-технического и бизнес-сотрудничества;
- описания центров-участников сети RTTN.

В администраторской части также представлены сервисы по сбору и формированию статистики и отчетов, а также сопровождению брокерами обращений к размещенным технологическим профилям и связанных с ними журналов сопровождения. Также предоставляются сервисы по отслеживанию появления новых членов сети с анализом количества и содержания размещенных ими профилей и статистики обращений по ним, сервисы мониторинга работы всех участников сети. Созданы сервисы по управлению новостями, относящимися как к какому-то конкретному региональному центру, так и ко всем пользователям сайта сети трансфера технологий в целом. Организована работа

библиотеки файлов с подборкой методических и презентационных материалов, наполняемой центрами и координирующей организацией сети.

После размещения в базе профили становятся доступными через интернет-сайт RTTN всем заинтересованным участникам трансфера технологий. Веб-сайт RTTN – это информационный ресурс, предназначенный для открытого предоставления посетителям информации о сети RTTN, центрах – ее участниках, технологических запросах и предложениях их клиентов на русском, английском и китайском языках. Открытость ресурса состоит в том, что незарегистрированные пользователи, также как и в сети EEN, имеют возможность поиска подходящих предложений или запросов через сайт. Просмотреть имеющиеся в базе технологические профили можно полным списком, или применив функцию поиска по фильтру.

Функция поиска технологических профилей через сайт осуществляется по названию искомой технологии, аннотации, описанию, инновационным аспектам или коду профиля. Также есть возможность ограничить область поиска, указав желаемый регион для поиска партнеров, выбрав конкретного участника сети, представившего профиль, или указав технологические ключевые слова, характеризующие разработку, и коды, характеризующие текущие или потенциальные области рыночного применения разработки.

с. Международная брокерская технологическая платформа Innoget

Недавно Российская сеть трансфера технологий установила партнерство с крупной международной брокерской технологической платформой Innoget [8], в рамках соглашения о сотрудничестве с которой реализован взаимный обмен технологическими запросами и предложениями из баз данных RTTN и Innoget. Таким образом, российские инновационные компании и разработчики, разместившие свои профили на платформе RTTN и заинтересованные в международной технологической кооперации, получили доступ к более чем 10 000 потенциальным зарубежным партнерам.

Технологическая биржа Innoget [8] была создана в 2007 г. и в настоящее время предоставляет тысячам специалистов и экспертов всего мира свободный доступ к технологическим запросам и предложениям, которые размещают международные компании и организации, заинтересованные в технологической и инновационной кооперации.

Принцип работы информационной платформы сети Innoget такой же, как у описанных выше сетей, отличия незначительны. Клиент сети имеет возможность найти интересующий его технологический профиль по ключевым словам (технологическим или рыночным) или по запросу. В результатах поиска приводится краткая информация о интересующем клиента технологическом профиле: наименования профиля, компания, предлагающая профиль, и список ключевых слов профиля. Детали технологического предложения или запроса описываются в формате, схожем с форматом профиля в сетях EEN и RTTN.

Из функциональных особенностей стоит отметить автоматический поиск похожих технологических профилей: список похожих профилей доступен клиенту сети в конце страницы с детальным описанием профиля. Регистрация на платформе Innoget доступна несколькими способами: созданием профиля Innoget или авторизацией через имеющиеся профили в сети facebook, google или LinkedIn. Зарегистрированные клиенты могут добавлять интересные профили в раздел «избранных», чтобы в дальнейшем автоматически получать обновления. Профиль также можно выгружать в формат pdf или делиться ссылкой на него посредством почтовых сервисов и социальных сетей.

3. Применение сетей трансфера технологий для оценки уровня технологизации результатов исследований

В последние годы в условиях замедления темпов экономического роста в России становится особенно актуальной задача оценки перспективности научно-технологических разработок и выделения тех технологий, которые в кратко- и среднесрочной перспективе могут обеспечить технологический прорыв и стать основой развития новых или традиционных индустрий [9].

Полученные в работе [9] данные дают основание говорить о том, что в последнее время промышленные компании становятся равноправными, а иногда и ключевыми участниками процесса генерации прорывного знания. В связи с этим при конкурсном отборе инициативных проектов для оказания им финансовой поддержки важно ориентироваться на уровень технологизации результата научной деятельности.

На текущий момент главными источниками информации при проведении научно-технической экспертизы по оценке уровня технологизации считаются патенты. Также в качестве информации для оценки уровня технологизации могут использоваться научные публикации, диссертации, авторефераты и другие данные. В данной работе сети трансфера технологий с размещенными в них технологическими запросами и технологическими предложениями рассматриваются как потенциальные источники для агрегации информации о результатах интеллектуальной деятельности, методах и технологиях, являющихся возможными аналогами результатов интеллектуальной деятельности, полученных или планируемых к получению в ходе выполнения прикладных научных исследований.

Для оценки уровня технологизации прикладных научных исследований предлагается использовать различные критерии, характеризующие уровень востребованности результатов НТД. Рассмотрим эти критерии и соответствующие им параметры, которые могут быть получены с применением информационно-аналитических систем, использующих специализированные методы и инструменты анализа больших коллекций документов.

Первый критерий – K_1 – это заинтересованность потребителей в результатах НТД, объем спроса на технологии, содержательно и тематически похожие на результаты НТД. Критерий может принимать значения «высокая», «средняя», «низкая» заинтересованность. Значение устанавливается экспертом с учетом предметной области на основе автоматически установленного значения параметра «Количество технологических запросов, содержательно и тематически похожих на результаты НТД».

Второй критерий – K_2 – это готовность производителей удовлетворить спрос на результаты НТД, объем предложения технологий, содержательно и тематически похожих на результаты НТД. Критерий может принимать значения «высокая», «средняя», «низкая» готовность. Значение устанавливается экспертом с учетом предметной области на основе автоматически установленного значения параметра «Количество технологических предложений, содержательно и тематически похожих на результаты НТД».

Эти два критерия показывают ситуацию, сложившуюся на технологическом рынке на текущий момент, но на их основе производится грубая оценка технологизации, так как не представляется возможным предсказать, как ситуация будет меняться в ближайшей перспективе.

Например, если значение отношения критерия K_1 к критерию K_2 значительно превышает единицу, т.е. спрос на технологию превышает предложение, то на первый

взгляд кажется, что новый результат НТД будет легко технологизируем. Однако, если посмотреть на ситуацию в динамике, может оказаться, что объем спроса на технологии на данный момент высокий, но интерес к теме стремительно падает. В такой ситуации необходимость в создании еще одного результата НТД по схожей тематике довольно низкая, и к моменту реализации планируемых разработок этот результат может оказаться невостребованным. Или другая ситуация, когда объем предложения на текущий момент низкий, но он стремительно растет в динамике. Тогда встает вопрос о том, смогут ли разработчики новой технологии выдержать конкуренцию, которая появится к моменту создания этой технологии.

Чтобы проанализировать подобные ситуации, необходимо провести анализ актуальности научно-технических разработок, т.е. анализ динамики тематически похожих результатов НТД во времени и потребности в этих результатах. Для этого предлагается использовать критерии K_3 и K_4 .

Критерий K_3 – это изменение во времени интереса потребителей технологий к тематике результата НТД, т.е. динамика спроса на технологии, содержательно и тематически похожи на ожидаемые результаты НТД. Критерий может принимать значения «растет», «постоянный», «падает».

Критерий K_4 – это изменение во времени интереса производителей технологий к тематике результата НТД, т.е. динамика предложения технологий, содержательно и тематически похожих на результаты НТД. Критерий может принимать значения «растет», «постоянный», «падает».

Критерии K_3 и K_4 устанавливаются на основе статистики количества, соответственно, технологических запросов и технологических предложений, содержательно и тематически похожих на результаты НТД, за выбранный диапазон времени. Графики, отображающие данную статистику, строятся автоматически при помощи информационно-аналитических систем.

На основе предлагаемых критериев можно сделать определенный вывод об актуальности ожидаемых результатов НТД. Окончательное решение об актуальности прикладных научных исследований предлагается принять эксперту с учетом их предметной области на основе полученных значений вышеназванных коэффициентов. Принимать решение об актуальности исследования стоит, опираясь на таблицу 1.

Таблица 1. Определение актуальности прикладных научных исследований на основе интереса к тематике со стороны потребителей и производителей новых технологий

Предложение \ Спрос	Падает	Постоянный	Растет
Падает	Не актуально	Не актуально	Не актуально
Постоянный	Актуально	Требуется более детальный анализ	Требуется более детальный анализ
Растет	Актуально	Актуально	Актуально

Для вычисления коэффициентов K_1 и K_3 необходим доступ к ресурсам, содержащим профили технологических запросов; для вычисления коэффициентов K_2 и K_4 необходим доступ к ресурсам, содержащим профили технологических предложений.

Для поиска тематически и содержательно похожих на тематику прикладных научных исследований технологических запросов и предложений предлагается использовать технологии интеллектуального анализа больших неструктурированных или слабо структурированных данных, реализованные

информационно-аналитической системе ExactusExpert [10, 11]. Для установления тематического сходства между документами проводится семантический анализ полных текстов документов. При этом тематическое сходство определяется на основе подсчета расстояния между векторами, состоящими из TF-IDF весов ключевых слов и словосочетаний. Из всех возможных словосочетаний выбираются именные словосочетания, вес которых превосходит эмпирически подобранный порог. Количество ключевых слов и словосочетаний определяется динамически в зависимости от размера документа и его типа. По всему набору слов и словосочетаний скользит окно, включающее фиксированный набор слов, и для каждого набора слов, покрываемых окном, вычисляется вес путем усреднения весов всех слов, входящих в окно. Если разница между весами двух смежных окон превосходит заранее выставленный порог, то окно сдвигается дальше по списку слов на фиксированную величину. Этот процесс позволяет выбрать для каждого документа достаточное количество ключевых слов и словосочетаний, позволяющих осуществить поиск тематически схожих документов независимо от их размера и типа [12].

Стоит отметить, что предлагаемый в данной работе набор критериев является достаточным для получения достоверной оценки уровня технологизации научных исследований. Помимо рассмотренных критериев следует принимать во внимание и другие критерии, учитывающие представленность результатов, содержательно и тематически похожих на результаты НТД, в разнотипных массивах научно-технических документов, включая патенты, авторефераты и диссертации, данные российских научных фондов и ресурсов-агрегаторов научного и научно-технического контента. Также желательно наравне с количественными характеристиками использовать критерии, опирающиеся на качественные параметры, дающие представление о степени проработанности предлагаемых научных разработок. Формируемая таким образом система критериев может послужить основой экспертной анкеты для оценки уровня технологизации результатов научно-технической деятельности и тем самым способствовать повышению обоснованности, объективности и достоверности экспертизы прикладных научных исследований.

4. Заключение

В статье проведен анализ функциональных возможностей ряда известных сетей трансфера технологий: Enterprise Europe Network, Russian Technology Transfer Network, Innoget. Анализ показал, что предоставляемые функции большинства сетей трансфера технологий ограничиваются поиском технологических запросов и/или технологических предложений по запросу, ключевым словам или кодам, характеризующим научно-техническую разработку и потенциальные области её применения.

Являясь связующим звеном между поставщиками технологий и их потребителями из различных областей знаний и НТД, сети трансфера технологий становятся источником ценной информации, с помощью которой можно оценивать востребованность и перспективность разработок, а также уровень их технологизации, что, в свою очередь, может использоваться при оценке потенциала коммерциализации результатов НТД.

В работе предложены критерии для оценки уровня технологизации прикладных научных исследований, полученные на основе анализа информации, хранящейся в сетях трансфера технологий. Значения критериев предлагается вычислять при помощи информационно-аналитической системы ExactusExpert, предоставляющей различные функции анализа текстовой информации, отсутствующие в сервисах, предоставляемых

сетями трансфера технологий, в частности, осуществлять поиск похожих документов. Рассматриваемые критерии дают представление о количественном соотношении технологических запросов и технологических предложений, тематически и содержательно похожих на оцениваемый результат НТД. Окончательное решение по актуальности и степени технологизации прикладных научных исследований предлагается принимать эксперту с учетом значений указанных критериев.

В качестве направлений дальнейших исследований стоит выделить разработку критериев, получаемых в результате обработки информации из других типов источников: патентных баз, коллекций диссертаций и авторефератов, научных публикаций и данных научных фондов. Причем разрабатываемые критерии должны характеризовать актуальность исследования не только посредством количественного аспекта, но и учитывать качество результата НТД. Дальнейшие исследования будут также направлены на проверку применимости предложенных в данной работе критериев для оценки уровня технологизации прикладных научных исследований.

Литература

1. John C. Mankins. Technology Readiness Levels: A White Paper [Электронный ресурс] // URL: <http://www.hq.nasa.gov/office/codeq/trl/trl.pdf>
2. Мартынов М.В., Сафонов А.А., Увайсов С.У. Сети трансфера технологий в мировом технологическом обществе // ИННОВАЦИИ. – 2012. – № 1. – С. 499-500.
3. Albors J., Sweeney E., Hidalgo A. Transnational technology transfer networks for SMEs. A review of the state-of-the art and an analysis of the European IRC network // Production Planning & Control. – 2005. – Т. 16. – № 4. – С. 413-423.
4. Enterprise Europe Network [Электронный ресурс] // URL: <http://een.ec.europa.eu>
5. Merlin. Client User Guide [Электронный ресурс] // URL: http://een.ec.europa.eu/tools/services/Help/OnlineHelp/Merlin_Client_UserGuide.pdf
6. Российская сеть трансфера технологий [Электронный ресурс] // URL: <http://rttn.ru/index.php/about-the-network>
7. Лукша О.П. и др. Российская сеть трансфера технологий, как составная часть национальной инновационной системы: методология, практика, перспективы / О.П. Лукша, А.Э. Яновский, Г.Б. Пильнов, Ю.Б. Алферов // Инновации. – 2009. – №10 (132). – С. 18-32.
8. Innoget [Электронный ресурс] // URL: <http://www.innoget.com>
9. Куракова Н., Зинов В., Комаров В., Павлов П. Долгосрочные прогнозы как инструмент формирования научно-технологической политики // Экономическая политика. – 2014. – № 4. – С. 7-32.
10. Тихомиров И.А. Exactus Expert: поисково-аналитическая система поддержки научно-технической деятельности / Тихомиров И.А., Смирнов И.В., Соченков И.В., Девяткин Д.А., Шелманов А.О., Зубарев Д.В., Швец А.В., Лешкин А.В., Суворов Р.Е. // Труды XIII национальной конференции по искусственному интеллекту с международным участием КИИ-2012, Белгород. – 2012. – Том 4. – С. 100 - 108.
11. Osipov G., Smirnov I., Tikhomirov I., Sochenkov I., Shelmanov A. Exactus Expert— Search and Analytical Engine for Research and Development Support // Novel Applications of Intelligent Systems. – Springer International Publishing. – 2016. – С. 269-285.
12. Суворов Р.Е., Соченков И.В. Определение связанности научно-технических документов на основе характеристики тематической значимости. // Искусственный интеллект и принятие решений. – №1. – 2013. – С.33-40.

STRIPS ПОСТАНОВКА ЗАДАЧИ ПЛАНИРОВАНИЯ ПОВЕДЕНИЯ В ЗНАКОВОЙ КАРТИНЕ МИРА

Киселёв Г.А. , Панов А.И.

Федеральный исследовательский центр “Информатика и управление” РАН
117312, Москва г, проспект 60-летия Октября, 9, kiselev@isa.ru

Аннотация

В работе представлены результаты экспериментов, проведенных с алгоритмом планирования поведения в знаковой картине мира в STRIPS постановке задачи. В картине мира алгоритм генерирует последовательность действий на основании компонентов элементов картины мира – компонентов знаков. Приведены основные шаги алгоритма планирования поведения - MAP алгоритма. Представлены основные компоненты программной реализации, проведены эксперименты с классической задачей STRIPS планирования “Мир блоков”.

1. Введение

Планирование – это составление последовательности действий, формируемых агентом на основе целей, информации об состоянии окружающей среды и динамике ее изменения. В качестве агента может выступать не только человек, но и интеллектуальная система, которая автоматически должна учитывать множество свойств среды: причины изменения среды, состояния среды, достаточность источников данных для получения информации о состоянии среды и характер изменения среды. Планирование интеллектуальной системой является интеллектуальным планированием. Одним из способов планирования является планирование в пространстве состояний. Такое планирование подразумевает рассмотрение в текущем состоянии тех действий, которые способствуют достижению цели. Обычно ситуацию представляют в виде набора фактов, являющихся замкнутыми атомарными формулами языка исчисления предикатов 1-го порядка. Среди всех состояний выделяют начальные состояния (обычно одно) и конечные (целевые). Проблема планирования в пространстве состояний состоит в поиске пути, ведущего из начального состояния в одно из конечных. В большинстве современных планировщиков задача планирования решается интеллектуальным агентом в терминах планировщика STRIPS[1].

Все действия агент применяет не откладывая, достигая целевые ситуации одну за другой по-отдельности. Пример описания среды в терминах STRIPS:

$$S = \{ \text{CLEAR}(a), \text{ONTABLE}(b), \forall x \forall y ((\text{CLEAR}(x) \wedge (x \neq y)) \rightarrow \neg \text{ON}(y, x)) \}$$

Здесь, CLEAR(a) означает, что «на блоке a пусто», ONTABLE(b) – «блок b лежит на столе», ON(a, b) – «блок a лежит на блоке b». Имена объектов из этого множества: ‘a’, ‘b’ – соответственно ‘блок a’, ‘блок b’. Выполнение агентом действий сводится к применению правил, которые модифицируют текущее (начальное) состояние для перевода его в другое (конечное).

В правиле должно быть описано 4 части: имя, условие, список добавлений и список удалений. Имя - это имя правила, условие – набор атомарных формул необходимых для применения этого действия, в список добавлений и в список удалений входят те атомарные формулы, которые описывают начальное и конечное состояния объектов и манипулятора, производящего действия.

В задачу STRIPS-планирования входят: домен, задача планирования и план. Домен - это описание действий, производимых в ситуации. Задача планирования – это набор из начальной и конечной ситуаций, решением данной задачи является построение плана, который достигает искомой цели. Планом являются наборы промежуточных состояний, наборы примененных правил и последовательность подстановок элементов предметной области вместо переменных так, чтобы цель была выполнима в последнем из промежуточных состояний.

Обычно используемое в задачах планирования представление знаний не позволяет решать задачи в коалиции агентов. Для этого в статье предлагается использовать знаковую модель картины мира для представления знаний агентов. В статье приведен алгоритм планирования в знаковой картине мира и представлен ряд экспериментов.

2. Планирование в картине мира

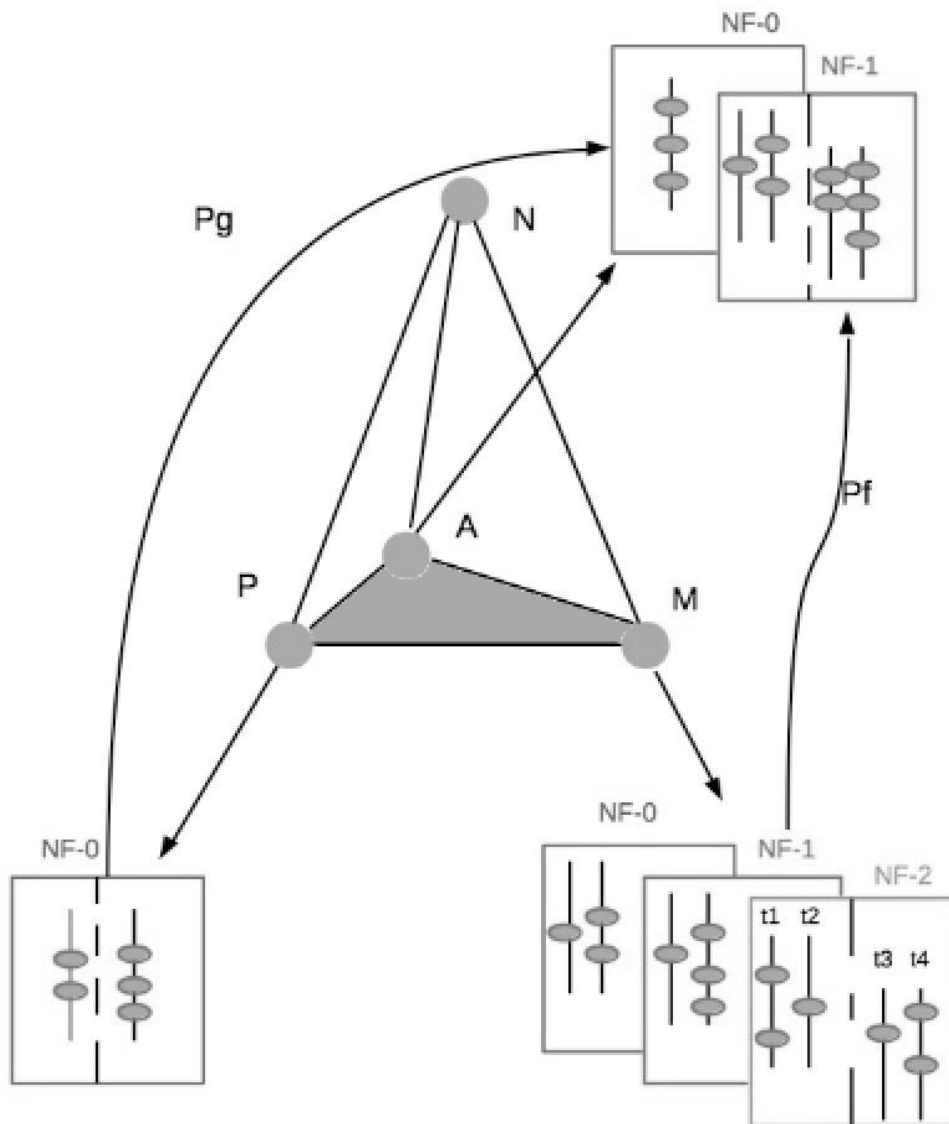


Рисунок 1 – Схема компонент знака

В настоящей работе используются понятия знака и знаковой картины мира, предложенные в [2, 3, 4]. Приведем краткое описание компонент знака по работе [2] (см. рис. 1). По А.Н. Леонтьеву [5] представление каждого предмета в сознании включает три компонента: образ предмета, его значение (или назначение) и личностные смыслы.

1. Понятие образа в психологии идентично понятию представления в семиотике: согласно понятию образа, развиваемому в когнитивной психологии, восприятие трактуется как процесс категоризации, что в точности соответствует понятию представления в семиотике – представление используется для различения объекта, соответствующего рассматриваемому знаку среди других объектов.

2. Личностные смыслы интерпретируются совокупностью действий, применимых субъектом действия к предмету [5]. В прикладной семиотике [6] это соответствует прагматической компоненте знака – множеству действий, связанных со знаком.

3. Понятие значения в психологии соответствует значению в семиотике и семантике, т.е. семантической компоненте знака.

Основной задачей данной статьи является описание алгоритма в терминах планировщика STRIPS постановки задачи для 1 агента. В статье рассматривается классический пример: “Мир блоков” (см. п. 4). В алгоритме задействованы различные элементы, такие как:

1. Фрагменты семантической сети на множествах компонент знаков.
2. Ситуации, представленные знаками.
3. Скрипты.
4. План решения, состоящий из последовательности действий для достижения целевой ситуации.
5. Знаки.
6. Действия.

Действия приводят к активации следующего фрагмента семантической сети. Фрагмент семантической сети состоит из левой и правой частей. Обе части каждого фрагмента состоят из набора ссылок на знаки, т. е. составляют списки условий и эффектов соответственно. На рис. 1 левая и правая части фрагментов семантической сети отделены пунктирной линией. Левая часть фрагмента семантической сети - это столбец условий, т. е. тех знаков, которые приводят к активации данного фрагмента семантической сети. Правая часть имеется только у фрагментов семантической сети, которые представляют действия. В ней находятся ссылки на знаки, которые являются эффектами опосредуемого действия. На примере знака “put-down” его фрагмент семантической сети будет выглядеть следующим образом:

Левая часть: {(Знак “holding”), (Знак “d”)}

Правая часть: {(Знак “d”), (Знак “clear”)}, {(Знак “ontable”), (Знак “d”)}, {(Знак “handempty”)}

В данной модели скриптами называются означенные действия, в которых производится подстановка объектов на места абстрактных ролей. Пример:

{(Знак “a”), (Знак “holding”)}, вместо {(Знак “block?x”), (Знак “holding”)}

Скрипты позволяют построить план решения задачи, путем перехода от одной промежуточной ситуации к другой до тех пор, пока не будет достигнута конечная ситуация задачи. В условиях действий в коалициях агентов, скрипты могут служить для связи между различными агентами и включать действия других агентов в личный план решения текущего агента.

Пример

Левая часть: {(Знак “agent1”), (Знак “field a”)}, {(Знак “barrier”), (Знак “near”)}, (Знак “agent2”), (Знак “afar”)}

Правая часть: {(Знак “agent1”), (Знак “field a”)}, {(Знак “barrier”), (Знак “near”)}, (Знак “agent2”), (Знак “field a”)}

Имя скрипта: call.

Подобные задачи составления общего плана решаются и другими мультиагентными алгоритмами. В качестве примера можно привести подход, описанный в работе [7], в которой говорится, что при составлении общего плана важен уровень взаимодействия. Чем выше уровень взаимодействия, тем чаще будут выбираться для решения задачи тех агентов, которые обладают низким уровнем производительности.

Перед построением плана поведения STRIPS задача планирования преобразуется в знаковое представление. При выполнении преобразования STRIPS задачи выстраиваются связи между списками различных объектов и их типами с учетом типов родительских объектов. Формируется задача, в которую входят списки фрагментов семантических сетей, составляющие значение, смысл и образ знака.

3. Описание MAP алгоритма

MAP алгоритм состоит из следующих 4 шагов:

1. В начале алгоритма задействуются начальная и целевая ситуации текущей задачи. До тех пор пока список знаков, входящих в смысл текущей ситуации не включен список знаков, которые входят в смысл начальной ситуации, выбираются все знаки, составляющие смысл текущей ситуации, они используются при генерации словаря скриптов этих знаков.
2. Если знаки, входящие в скрипты, не опосредуют собой действие и у них не пустой столбец условий (существуют знаки, которые находятся в столбце условий), генерируется словарь фрагментов, где фрагменты смыслов заполняются по фрагменту сети на образах знаков.
3. После этого удаляются те фрагменты сети смыслов данного знака, которые повторяются. Создается новый фрагмент сети смыслов, в котором в столбец условий добавляются значения эффектов предыдущего фрагмента и создается ситуация, где имеются 3 составляющие: номер фрагмента семантической сети, его имя и скрипт.
4. Ситуация добавляется в план решения задачи, цикл переходит к следующему фрагменту сети.

4. Эксперименты

Разработана программная система, реализующая MAP алгоритм на языке Python. Диаграмма классов программной реализации MAP алгоритма приведена на рис. 2.

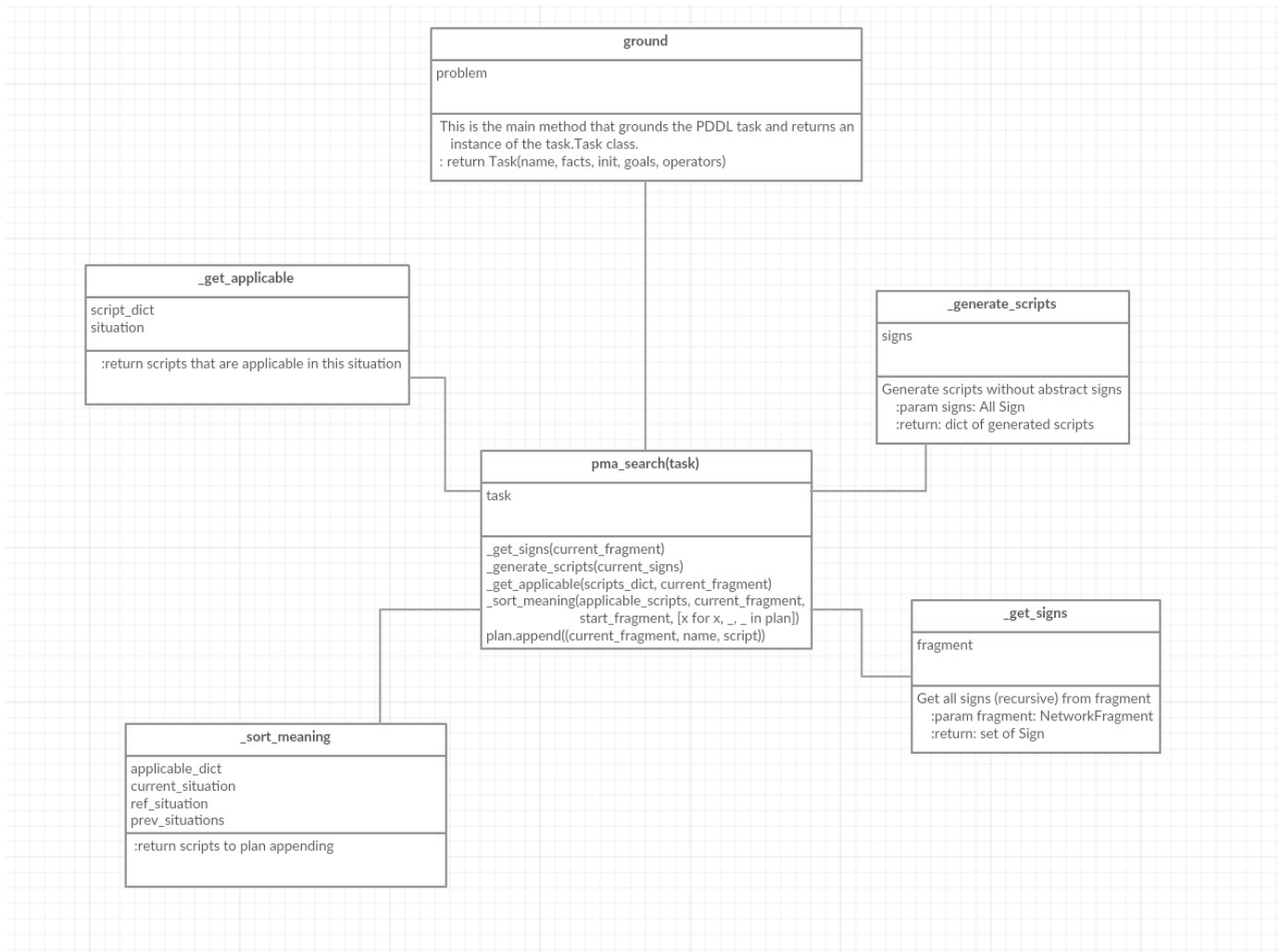


Рисунок 2 - Диаграмма классов программной реализации MAP алгоритма

Проведены эксперименты на классической задаче по планированию действий “Мир блоков” с различными начальными и конечными ситуациями. Ситуации описаны в терминах STRIPS постановки задачи:

Задача 1:

(define (problem BLOCKS-4-0)

(:domain BLOCKS)

(:objects D B A C - block)

(:INIT (CLEAR C) (CLEAR A) (CLEAR B) (CLEAR D) (ONTABLE C) (ONTABLE A)

(ONTABLE B) (ONTABLE D) (HANDEEMPTY))

(:goal (AND (ON D C) (ON C B) (ON B A)))

)

Начальная ситуация: 4 блока лежат на столе, на блоках ничего нет, манипулятор пустой.

Конечная ситуация: блок d на блок c, c на b, b на a.

Задача 2:

(define (problem BLOCKS-4-1)

(:domain BLOCKS)

(:objects A C D B - block)

(:INIT (CLEAR B) (ONTABLE D) (ON B C) (ON C A) (ON A D) (HANDEEMPTY))

(:goal (AND (ON D C) (ON C A) (ON A B))))

Начальная ситуация: на блоке b ничего нет, d на c, c на a, a на b.

Конечная ситуация: блок d на блок c, c на b, b на a.

Часть домена задачи:

(:action pick-up

:parameters (?x - block)

:precondition (and (clear ?x) (ontable ?x) (handempty))

:effect

(and (not (ontable ?x))

(not (clear ?x))

(not (handempty))

(holding ?x)))

В ходе построения плана алгоритм создает фрагменты семантической сети на компонентах смыслов знаков. В начале выполнения алгоритма фрагменты семантической сети значений и образов знаков отображены на рис. 3. На рисунке 3а изображен фрагмент семантической сети значений знака, на рисунке 3б изображен в фрагмент семантической сети образов знака.

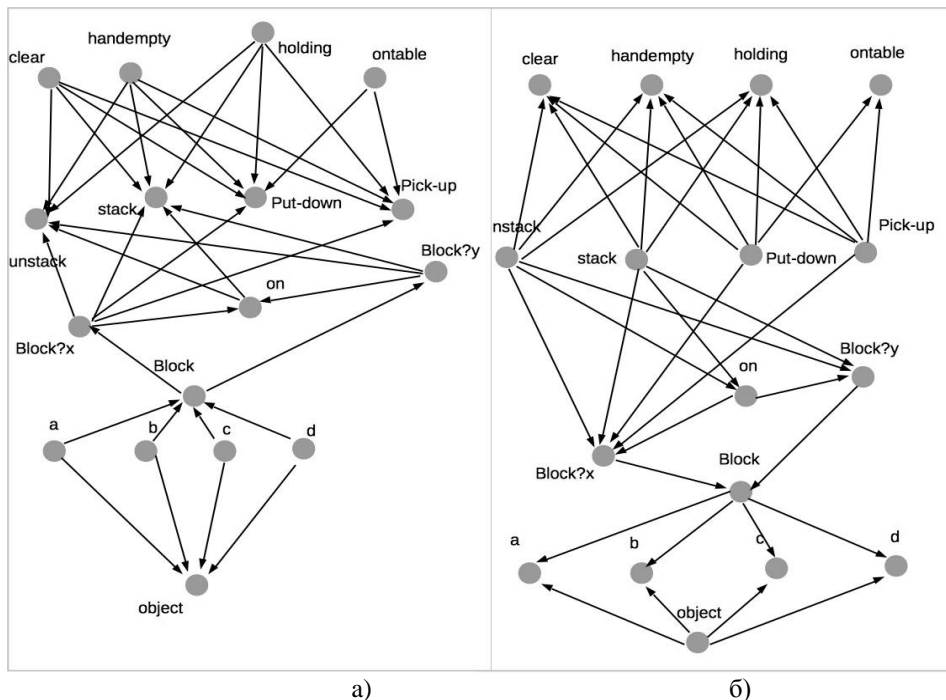


Рисунок 3 - Фрагменты семантической сети на множестве значений и образов знаков

На рис. 4 изображены фрагменты семантической сети на множестве личностных смыслов знаков. На рис 4а – сеть на множестве смыслов в начальной ситуации, на рисунке 4б – сеть на множестве смыслов после применения первого действия. Толщина стрелок определяет количество созданных фрагментов семантической сети на множестве личностных смыслов знаков.

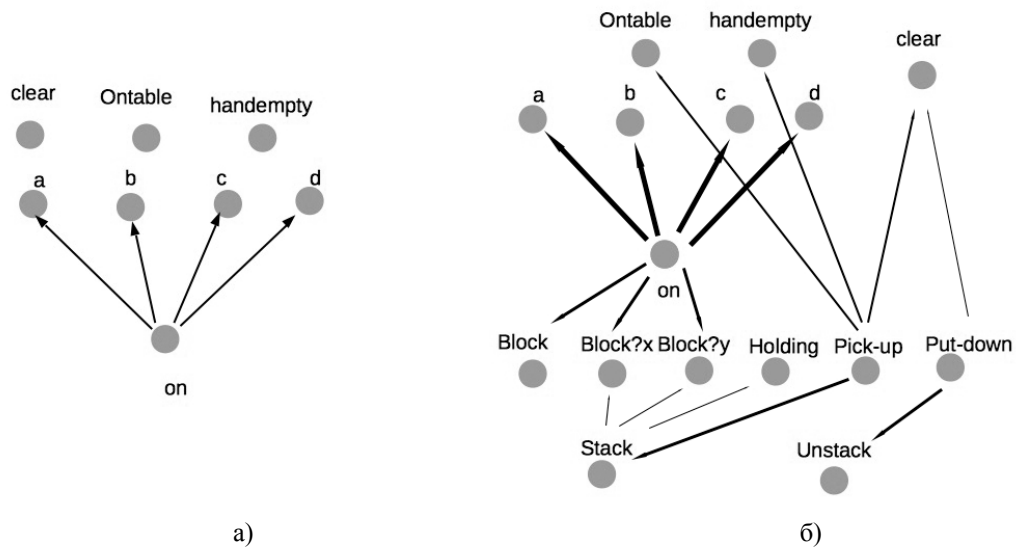


Рисунок 4 – Фрагменты семантической сети личностных смыслов знаков

С первой задачей алгоритм справился за 14 шагов и создал 17 знаков. Со второй задачей алгоритм справился за 8 шагов и создал 17 знаков. Также задачи были поданы на вход планировщиков, использующих эвристики A^* и W^* . Сравнение количества шагов приведено на рис. 5. Из рисунка видно, что на рассмотренных задачах MAP показывает такие же результаты, что и известные алгоритмы планирования. В дальнейшем планируется проведение большего количества экспериментов для демонстрации преимущества MAP-планировщика.

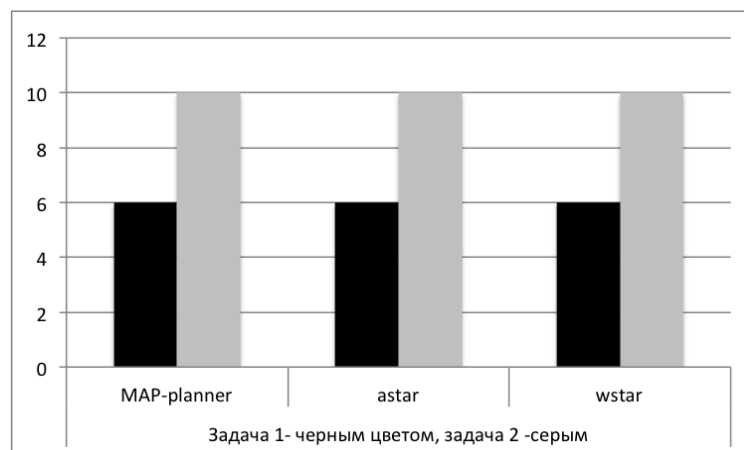


Рисунок 5 – Гистограмма длины сгенерированных планов для 2 задач “Мира блоков”

5. Заключение

В статье рассмотрена программная реализация алгоритма планирования поведения в знаковой картине мира. Были проведены эксперименты с программной реализацией алгоритма, проведено сравнение с известными алгоритмами, использующими A^* и W^* в качестве эвристик. В дальнейшем будет добавлена возможность согласованности планов различных агентов между собой, выработка общего плана для множества агентов и проведен ряд новых экспериментов.

Работа выполнена при финансовой поддержке РФФИ, проект №15-07-06214.

Литература

1. Fikes R.E., Nilsson N.J. STRIPS: A new approach to the application of theorem proving to problem solving // Artif. Intell. 1971. Vol. 2, № 3-4. P. 189–208.
2. Осипов Г.С., Панов А.И., Чудова Н.В. Управление поведением как функция сознания. I. Картина мира и целеполагание // Известия Российской Академии Наук. Теория и системы управления. 2014. № 4. P. 49–62.
3. Осипов Г.С., Панов А.И., Чудова Н.В. Управление поведением как функция сознания. II. Синтез плана поведения // Известия РАН. Теория и системы управления. 2015. № 6. P. 47–61.
4. Чудова Н.В. Концептуальная модель картины мира для задачи моделирования поведения, основанного на сознании // Искусственный интеллект и принятие решений. 2012. № 2. P. 51–62.
5. Леонтьев А.Н. Деятельность. Сознание. Личность. Изд. 2-е. М.: Политиздат, 1977. 304 p.
6. Поспелов Д.А. Прикладная семиотика и искусственный интеллект // Программные продукты и системы. 1996. № 3. P. 10–13.
7. Hexmoor H., Eluru S., Sabaa H. Plan sharing: Showcasing coordinated UAV formation flight // Inform. 2006. Vol. 30, № 2. P. 183–192.

ДИСКУРСИВНЫЙ АНАЛИЗ В ЗАДАЧАХ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

Ананьева М.И.

Институт системного анализа ФИЦ ИУ РАН
117312, пр-т 60-летия Октября 9, Москва, ananyeva@isa.ru

Кобозева М.В.

Институт системного анализа ФИЦ ИУ РАН
117312, пр-т 60-летия Октября 9, Москва, mvharitonova@yandex.ru

Аннотация

В докладе поднимается вопрос о роли дискурсивного анализа в области обработки текстов на естественном языке. Авторами представлен обзор существующих дискурсивных парсеров, а также последних исследований по применению дискурсивного анализа для повышения качества машинного перевода, автоматического реферирования, определения связности текста и вопросно-ответного поиска. Авторы приходят к выводу, что решение многих задач автоматической обработки естественного языка невозможно без учета дискурсивной структуры текста. Кроме того, рассматривается возможность применения дискурсивно анализа в рамках Теории риторических структур для русского языка.

1. Введение

Для решения различных задач обработки языка в современной компьютерной лингвистике активно применяется морфологический, синтаксический и семантический анализ. Однако текст представляет собой не просто линейную последовательность предложений, а иерархическую структуру: клаузы объединяются в предложе-

ния, предложения в абзацы и т.д. Большую роль играют такие факторы, как позиция предложений, их порядок, контекст и окружение [1]. Поэтому дискурсивный анализ текста, выходящий за рамки предложения, существенно повышает качество решения многих задач в этой области. На сегодняшний день в исследованиях, посвящённых обработке текстов на русском языке, анализу дискурсивной структуры уделяется недостаточное внимание: пока не существует ни русскоязычных корпусов с соответствующей разметкой, ни тем более автоматических дискурсивных парсеров. В данной статье представлено обоснование необходимости использования дискурсивного анализа во многих задачах компьютерной лингвистики, а также описывается создание русскоязычного корпуса текстов с дискурсивной разметкой.

2. Модели дискурса

Теоретические исследования в области анализа дискурса и связности текста ведутся достаточно давно, а именно со второй половины 1970-х годов. Тогда особое внимание уделялось распознаванию глубинной структуры текста на основе лингвистического анализа, оперирующего понятиями элемента дискурса и отношениями между этими элементами [1-4]. Однако исследования дискурсивной структуры актуальны и до сих пор, о чем свидетельствуют проводимые в настоящее время открытые соревнования автоматических дискурсивных парсеров, например, CoNLL-2015 [5]. Работы последних лет, связанные с автоматическим анализом дискурсивной структуры текста, можно разделить на две группы: одни посвящены построению моделей дискурсивной структуры текста и методам её автоматического извлечения, другие - использованию результатов анализа дискурса для улучшения качества решения других задач автоматической обработки текстов на естественном языке.

К основным моделям дискурсивной структуры текста, которые используются при разработке автоматических парсеров, можно отнести: Теорию риторических структур (Rhetorical Structure Theory) [6], Теорию общей структуры документов (Cross-document Structure Theory) [7], модель Penn Discourse Treebank [8], Теорию сегментной репрезентации дискурса (Segmented Discourse Representation Theory) [9], Теорию представления дискурса (Discourse Representation Theory) [10].

Более подробно мы остановимся на наиболее известной модели дискурса – Теории риторических структур (TRC) Манна и Томпсон, 1988 [6]. Согласно этой теории, иерархическая структура связного текста устроена следующим образом. Каждая элементарная единица дискурса (в своем базовом варианте – клауза) связана риторическим (или дискурсивным) отношением по крайней мере с одной другой единицей и является узлом дискурсивного дерева. Внутри отношения дискурсивная единица может выступать в роли ядра (основной части) или сателлита (сопутствующей части). В зависимости от распределения этих ролей отношения делятся на ассиметричные (одноядерные) и симметричные (мультиядерные). Так как авторы TRC оставляют другим исследователям возможность варьировать список отношений, в разных работах он может существенно различаться (количество отношений может достигать 80-ти), однако базовый набор включает 23 отношения. Приведем некоторые из них: одноядерные – причина, следствие, антитеза, детализация, оценка, обстоятельство, резюме и др., мультиядерные – контраст, сравнение, переформулировка и др. В TRC риторические отношения определяются на основе функциональных и семантических признаков, поэтому наличие формальных признаков (лексических маркеров) для определения того или иного отношения факультативно.

В лаборатории "Компьютерной лингвистики и интеллектуального анализа информации" ИСА ФИЦ ИУ РАН проводится адаптация Теории риторических структур для русского языка, а также ведется работа по созданию анализатора дискурсивной структуры текста. ТРС является универсальной моделью с точки зрения применимости к разным языкам, поэтому авторы статьи и решили опереться на нее в своем исследовании.

Первая задача, которая встает при создании автоматического парсера, – это создание корпуса текстов с соответствующей разметкой. На данном этапе нами разработана инструкция по дискурсивной разметке, описывающая принципы выделения элементарных дискурсивных единиц, а также правила выбора тех или иных риторических отношений между ними. В ходе разметки мы уточнили список риторических отношений:

Одноядерные отношения:

1. Background (Фон)
2. Evidence (Обоснование)
3. Cause (Причина)
4. Effect (Следствие)
5. Condition (Условие)
6. Purpose (Цель)
7. Concession (Уступка)
8. Preparation (Подготовка)
9. Conclusion (Вывод)
10. Elaboration (Детализация)
11. Antithesis (Антитезис)
12. Solutionhood (Решение)
13. Motivation (Мотивация)
14. Evaluation (Оценка)
15. Attribution (Источник)

Мультиядерные отношения:

1. Contrast (Контраст)
2. Restatement (Переформулировка)
3. Sequence (Последовательность)
4. Joint (Кнъюнкция)
5. Comparison (Сравнение)
6. Same-unit (Прерывающаяся единица)

В качестве материала для корпуса были выбраны 30 текстов различных жанров из корпуса СинТагРус. Для построения риторических деревьев используется система руч-

ной разметки текстов с открытым исходным кодом, разработанная А.Зельдесом². На рисунке 1 приведен пример разметки нескольких предложений.

Кроме того, в рамках данного исследования ведется работа по созданию списка лексических маркеров риторических отношений (на данный момент рассматривается около 50 маркеров), а также по выявлению соотношения этих маркеров с конкретными риторическими отношениями на основе размеченных текстов.

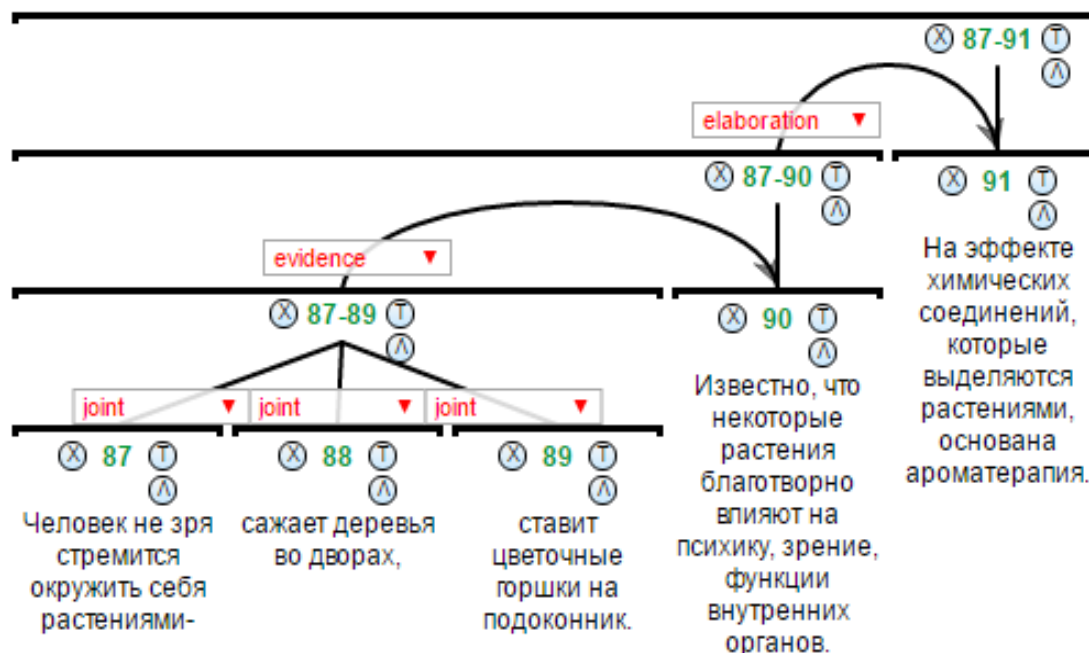


Рис. 1. – Пример разметки текста

3. Автоматические парсеры на базе Теории риторических структур

На сегодняшний день в рамках Теории риторических структур создано большое количество корпусов текстов с дискурсивной разметкой для разных языков (английский, немецкий, испанский, португальский, японский, китайский и др.). Кроме того, существует довольно большое число автоматических дискурсивных парсеров. В целом задача таких парсеров заключается в том, чтобы определить, как дискурсивные элементы соединены между собой, и на основе этого построить иерархическую структуру текста.

Один из первых алгоритмов дискурсивной разметки был разработан Марку [11] для английского языка. Данный алгоритм основан на правилах, в первую очередь учитывающих лексические дискурсивные маркеры как показатели риторических отношений. Для каждого маркера существует правило, которое присваивает ему определенные характеристики: риторическое отношение, с которым он коррелирует; порядок следования ядра и сателлита в паре дискурсивных единиц; длина дискурсивных единиц. В алгоритме также учитывается корреляция между риторической структурой текста и границами его предложений, абзацев и глав. Предложенный Марку алгоритм отработал на 84,9% (по F1-мере) для определения маркеров дискурсивных отношений и на 85,5% (по F1-мере) для определения границ клауз.

²<https://corpling.uis.georgetown.edu/rstweb/info/>

В парсере RASTA (The Rhetorical Structure Theory Analyzer), созданном Корстон-Оливером [12], в качестве показателей риторических отношений помимо лексических маркеров используются синтаксические характеристики дискурсивных единиц для разрешения неоднозначности лексических маркеров. Для всех риторических отношений автор определяет набор обязательных синтаксических критериев, например: состоят ли элементы отношения в сочинительной или подчинительной связи; порядок следования элементов и др.

Риторический парсер Райтера [13] работает на основе машинного обучения и тестировался как для английского, так и для немецкого языка. Кроме лексических маркеров учитывается: какой артикль вводит именную группу (определенный или неопределенный), знаки пунктуации, лексическая близость слов, а также длина дискурсивных единиц. При анализе в качестве решающей функции используется метод опорных векторов. Аккуратность (т.е. отношение правильно классифицированных системой примеров к общему количеству примеров) для английского языка составляет 61,8%, для немецкого – 39,1%.

Существуют работы по развитию алгоритмов, предложенных Марку и Корстон-Оливером. Так, в работе [14] расширяется понятие лексических маркеров: авторы предлагают учитывать специальные слова в главной именной группе (в подлежащем) и глагольной группе (в сказуемом). Например, в предложении [*New York style pizza meets California ingredients,*] [*and the result is the pizza from this Church Street pizzeria.*] существительное “*result*” маркирует отношения следствия. Кроме того, существуют исследования, в которых алгоритм Марку применяется к другим языкам. Сюда можно отнести риторический парсер DiZer [15], разработанный для бразильского и португальского.

Перечисленные работы в основном посвящены автоматическому анализу дискурса на уровне предложений. Однако более сложной задачей является переход на уровень текста, так как в этом случае необходимо учитывать коммуникативную функцию каждого элемента, а лексические и синтаксические маркеры уже не играют роли. Поэтому в более поздних исследованиях применяется глубокое обучение. Кроме того, в последние годы основной тенденцией в области разработки дискурсивных парсеров стало комбинирование в алгоритме двух классификаторов: классификатора сочетаемости, который определяет, образуют ли смежные дискурсивные единицы поддерево, и классификатора отношений, который вслед за первым определяет, какой тип отношений установить для нового поддерева.

Такой алгоритм, например, используется в парсере Hilda [16], основанном на методе опорных векторов и машинном обучении на корпусе RST-DT. F1-мера построения дискурсивной структуры данным парсером составляет 75,8%. Hilda получает развитие в дальнейших исследованиях [17], где авторы используют дополнительные лингвистические характеристики, такие как семантическая близость, правила порождения дискурса и др. F1-мера построения дискурсивной структуры данным парсером составляет 89,51%. В одной из последних работ по автоматическому анализу дискурса [18] предложен алгоритм парсинга, основанный на рекурсивной нейронной сети. Также следует упомянуть парсер Codra [19], алгоритм которого заключается в применении двух условных случайных полей: одно для анализа на уровне предложений, другое для анализа на уровне текста. Выделение дискурсивных единиц, определение ядра и установление связей на уровне предложений Codra выполняет лучше метода, выбранного авторами статьи в качестве базового, на 5,4%, 17,6% и 12,8%, соответственно (F1), а на уровне текста - на 5,7%, 6,4% и 9,5%.

Итак, на сегодняшний день задача выделения клауз и установления отношений между ними решается с достаточно высоким качеством, что позволяет использовать дискурсивный анализ для решения многих практических задач, о которых пойдет речь ниже.

4. Использование моделей дискурса в задачах обработки естественного языка

Работы, ориентированные на практическое применение результатов дискурсивного анализа, описывают решение таких задач как машинный перевод, автоматическое реферирование отдельных текстов, автоматическое реферирование коллекции текстов, анализ тональности, проверка связности текста, жанровая классификация, определение авторства и другие. Рассмотрим некоторые из них подробнее.

Машинный перевод – одна из наиболее сложных задач обработки естественного языка, в которой сильна потребность в модели структуры текста, объединяющей простые предложения между собой. Так, в диссертации [20] рассматривается применение дискурсивных связей для улучшения качества работы фразовой статистической модели машинного перевода. Основная идея заключается в объединении всего документа в единую структуру и переводе как единого объекта, в отличие от более традиционной схемы, при которой фразы переводятся по отдельности. С лингвистической точки зрения наибольшее внимание уделяется местоименной (прономинальной) анафоре. Автором предложен вычислительно эффективный алгоритм, а также платформа для проведения экспериментов по влиянию дискурса на качество машинного перевода, но конкретных оценок, характеризующих масштаб улучшения, не приведено.

Авторы работы [21] предложили критерии подобия на основе TPC и экспериментальным путем показали, что такой подход улучшает корреляцию (поСпирмену) между списками вариантов переводов, отранжированными системой и экспертами, на 2,5% - 3,5% в зависимости от корпуса и варианта алгоритма. А в своей следующей работе [22] настройкой параметров алгоритма авторы добились увеличения коэффициента корреляции Спирмена на 9,5%.

В работе [23] предложены эксперименты, показывающие, что аккуратность перевода логических связок с учетом частей речи и дискурсивных меток на 1,4% лучше учета только дискурсивной структуры, на 5,7% лучше варианта, опирающегося только на части речи, и на 8,5% лучше перевода текста без меток.

Эксперименты, доказывающие пользу дискурса для разных задач автоматической обработки естественного языка, проводит и исследователь из Швейцарии Т.Майер (ThomasMeyer). На примере английского-чешского параллельного корпуса он показывает, в каких случаях знание дискурсивной структуры текста более важно для качества машинного перевода, а в каких – менее [24]. Во многих вышеописанных работах определение дискурсивного отношения происходит по связующим словам (союзам, наречиям...) и порядку предложений в тексте. Однако такой подход работает не в ста процентах случаев. Так, в другой работе Т. Майера [25] показано, что человек при переводе не использует равнозначные оригиналу связующие слова в 18% случаев, а в системах машинного перевода это происходит только в 7% случаев. Эти результаты говорят о необходимости учета семантических и функциональных признаков, а не только лексических маркеров, при машинном переводе.

Другой задачей, для решения которой уже доказана важность учета дискурсивной структуры текста, является автоматическое реферирование. Объектом реферирования может выступать как отдельный текст [26], так и целая коллекция документов [27], когда пользователя интересует не общий смысл отдельно взятой работы, а информация о некотором предмете или явлении, собранная из разных источников. В работе по реферированию отдельного документа [26] предлагается решение проблемы об оптимальной глубине ветвления дискурсивной структуры³. На базе этих идей авторы разработали свой дискурсивный парсер, который позволяет решить задачу реферирования с F1-мерой на 5,3% лучше базового метода. Во второй работе (реферирование коллекции документов) авторы комбинируют две дискурсивные модели – Теорию риторических структур (RST) и Теорию общей структуры документов (CST), применяя их вместе, отдельно, с нормализацией и без, с учетом подразделов текста и т.д. Комбинация обеих моделей дискурса показала лучшую F1-меру (44,19%), немного хуже отработала комбинация CST и RST с нормализацией по высоте дерева (F1-мера 43,91%).

Другой коллектив применял дискурсивный анализ для генерации рефератов отзывов о продуктах и заведениях [28]. При этом результаты работы всех участвующих в эксперименте систем оценивались двумя группами экспертов: первая не была знакома с полными текстами отзывов, а вторая группа все отзывы прочитала. По результатам оценки автоматически составленных рефератов первой группой экспертов качество работы системы, разработанной авторами, достигло 71% (что на 22% лучше результата других систем), а во втором случае - 69% (что на 6% лучше результата других систем).

Дискурсивный анализ также успешно применяется в области анализа тональности. Тональность текста чаще всего классифицируется на позитивную или негативную, или оценивается по шкале от одного до пяти. В работе [29] показано, как учет дискурсивной структуры текста улучшает результат автоматического анализа тональности на 4%, а у Хиршопа аккуратность определения полярности сообщения увеличилась на 4,5% [30].

Фенг и коллеги в своей работе [31] представляют исследование применения дискурсивного анализа для решения задачи определения связности текста (т.е. наличия логических связей между предложениями, тема-рематических цепочек), увеличив аккуратность на 4,2% на уровне предложений и на 21,3% на уровне целого текста. В диссертации [32], помимо определения связности текста, Фенг применяет дискурсивный анализ для идентификации авторства на основе чанков, которая с учетом дискурсивных структур улучшилась на 3% по сравнению с базовым методом. В работах [33-34] того же автора приводится описание способа применения дискурсивной структуры для обогащения признакового пространства при определении авторства текстов на английском, испанском и греческом языках. Для первых двух языков удалось добиться 100% аккуратности, тогда как для греческого - только 40%.

Дискурсивный анализ используется и при разработке вопросно-ответных систем. В работе [35] авторы рассматривают возможность учета дискурса при поиске ответа на цепочку семантически близких вопросов. В этом случае целью анализа дискурса является автоматическое определение дискурсивных отношений между вопросами. При помощи синтаксического, семантического и дискурсивного анализа авторы предлагают

³Существует два подхода к построению дискурсивных деревьев: объединение элементарных дискурсивных единиц (обычно - клауз), являющихся узлами риторической сети, в группы по близости с установлением отношений (симметричных или асимметричных) без объединения таких групп друг с другом (так называемая поверхностная дискурсивная структура), и построение иерархической структуры, в которой такие группы объединяются друг с другом (глубокая дискурсивная структура).

составлять когнитивную карту запроса пользователя, которая помогает совершенствовать интерпретацию вопросов и поиск ответа. В работе [36] дискурсивный анализ применяется для усовершенствования системы ответов на категориальные вопросы. Ответы на такие вопросы никогда не представляют собой простых именных групп, а содержат несколько предложений, которые включают такие риторические отношения как Причина, Мотивация, Цель и Объяснение. Система дает правильные ответы в 60% случаев. В исследовании [37] разрабатывается алгоритм ответов на вопросы, состоящих из нескольких предложений, который включает в себя извлечение риторической структуры вопросов.

Из отечественных исследователей, работающих в области дискурсивного анализа, стоит отметить А.А. Кибрика. В одной из своих работ [38] Кибрик и его коллеги предложили модель референциального выбора с учетом дискурсивной структуры текста. Ранее этим коллективом было доказано, что наиболее важным фактором для решения данной задачи является расстояние от клаузыреферента, до клаузы антецедента, которое измеряется числом шагов по риторическому дереву. За основу был взят уже размеченный корпус RST-DT, к которому для данного исследования была добавлена анафорическая разметка. Кибрику и его коллегам удалось достичь уровня аккуратности предсказания на 14,8% выше, чем у аналогичных систем.

Под руководством А.А. Кибрика совместно с медиками проводилось также исследование рассказов детей о сновидениях, большая часть которых страдала невротическими заболеваниями [39]. Исследователи выявили различия в структуре рассказа у детей с неврозами и без. Так, было отмечено, что дети с диагнозом строят более длинные клаузы, а также гораздо чаще используют отношения *контраст*, *причина*, *результат*, *переформулировка* и *фон*. А кроме того, их рассказы имеют более сложную ветвящуюся дискурсивную структуру. Такое исследование доказывает принципиальную возможность адаптации ТРС для русского языка, хотя и совершенно не подходит для решения задач компьютерной лингвистики. Группа А.А. Кибрика модифицировала оригинальный список отношений, предложенный Манном и Томпсон, дополнив его 20 новыми, однако 17 из них описывают коммуникативные связи (*вопрос*, *ответ*, *возражение*...) и оформительные компоненты дискурса (*концовка*, *отступление* и др.), что не столь важно в письменных текстах.

Обзор работ в области применения дискурсивного анализа при решении задач обработки естественного языка показывает, что в большинстве случаев учет дискурсивной структуры текста повышает качество решения подобных задач. Однако, на сегодняшний день даже многие отечественные ученые предпочитают работать не с русским, а с английским языком. Это объясняется наличием готовых автоматических парсеров и размеченных корпусов для английского языка, создание которых с нуля – очень трудо- и времязатратная задача.

5. Заключение

В настоящей работе описано современное состояние исследований по автоматическому выделению дискурсивной структуры текста. Показано, что дискурсивный анализ позволяет эффективно решать многие задачи обработки текстов на естественном языке. В то время как для английского существует довольно большое количество наработок в данной области, для русского языка таких исследований крайне мало. На данный момент не существует ни готовых дискурсивных парсеров, ни размеченных корпусов, на которых можно было бы обучить и протестировать систему для русского языка. Однако, авторами статьи ведется работа по адаптации Теории риторических структур для письменных русскоязычных текстов разных жанров и созданию вручную размеченного корпуса.

Литература

1. Webber B., Egg M., Kordoni V. Discourse structure and language technology // *Natural Language Engineering*. – 2012. – Т. 18. – №. 04. – С. 437-490.
2. Linde C., Goguen J. A. Structure of planning discourse // *Journal of Social and Biological Structures*. – 1978. – Т. 1. – №. 3. – С. 219-251.
3. Grosz B. J., Joshi A. K., Weinstein S. Providing a unified account of definite noun phrases in discourse // *Proceedings of the 21st annual meeting on Association for Computational Linguistics*. – Association for Computational Linguistics, 1983. – С. 44-50.
4. Longacre R., Levinsohn S. Field analysis of discourse // *Current trends in text linguistics*. – 1978. – Т. 103. – С. 22.
5. Xue N. et al. The conll-2015 shared task on shallow discourse parsing // *Proceedings of CoNLL*. – 2015. – С. 2.
6. Mann W. C., Thompson S. A. Rhetorical structure theory: Toward a functional theory of text organization // *Text-Interdisciplinary Journal for the Study of Discourse*. – 1988. – Т. 8. – №. 3. – С. 243-281.
7. Radev D. R. A common theory of information fusion from multiple text sources step one: cross-document structure // *Proceedings of the 1st SIGdial workshop on Discourse and dialogue-Volume 10*. – Association for Computational Linguistics, 2000. – С. 74-83.
8. Miltsakaki E. et al. The Penn Discourse Treebank // *LREC*. – 2004.
9. Lascarides A., Asher N. Segmented discourse representation theory: Dynamic semantics with discourse structure // *Computing meaning*. – Springer Netherlands, 2008. – С. 87-124.
10. Kamp H., Van Genabith J., Reyle U. Discourse representation theory // *Handbook of philosophical logic*. – Springer Netherlands, 2011. – С. 125-394.
11. Marcu D. The rhetorical parsing of natural language texts // *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. – Association for Computational Linguistics, 1997. – С. 96-103.
12. Corston-Oliver S. H., Corston-Oliver S. H. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis // *The AAAI Spring Symposium on Intelligent Text Summarization*. – 1998. – С. 9-15.
13. Reitter D. Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models // *LDV Forum*. – 2003. – Т. 18. – №. 1/2. – С. 38-52.
14. Le H., Abeysinghe G. A Study to Improve the Efficiency of a Discourse Parsing System // *Computational Linguistics and Intelligent Text Processing*. – 2003. – С. 101-114.
15. Pardo, T.A.S., Nunes, M.G.V. 2006. Review and Evaluation of DiZer - An Automatic Discourse Analyzer for Brazilian Portuguese. *PROPOR 2006, Lecture Notes in Computer Science*. 3960, Springer-Verlag, Berlin Heidelberg New York, 180-189.
16. Hernault H, Prendinger H, DuVerle DA, Ishizuka M. HILDA: a discourse parser using support vector machine classification." *Dialogue and Discourse* 1.3 (2010): 1-33.
17. Feng, Vanessa Wei, and Graeme Hirst. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 60-68. Association for Computational Linguistics, 2012.

18. Li J., Li R., Hovy E. H. Recursive Deep Models for Discourse Parsing //EMNLP. – 2014. – C. 2061-2069.
19. Joty S., Carenini G., Ng R. T. CODRA: A novel discriminative framework for rhetorical analysis //Computational Linguistics. – 2015.
20. Hardmeier C. Discourse in statistical machine translation. – 2014.
21. Guzmán, F., Joty, S., Màrquez, L., & Nakov, P. Using Discourse Structure Improves Machine Translation Evaluation //ACL (1). – 2014. – C. 687-698.
22. Joty S., Nakov P. DiscoTK: Using discourse structure for machine translation evaluation //In Proceedings of the Ninth Workshop on Statistical Machine Translation. – 2014.
23. Meyer, T., Popescu-Belis, A., Hajlaoui, N., & Gesmundo, A. Machine translation of labeled discourse connectives //Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA). – 2012. – №. EPFL-CONF-192524.
24. Meyer T., Poláková L. Machine translation with many manually labeled discourse connectives //Proceedings of the 1st DiscoMT Workshop at ACL 2013 (51st Annual Meeting of the Association for Computational Linguistics). – 2013. – №. EPFL-CONF-192526.
25. Meyer T., Webber B. Implication of discourse connectives in (machine) translation //Proceedings of the 1st DiscoMT Workshop at ACL 2013 (51st Annual Meeting of the Association for Computational Linguistics). – 2013. – №. EPFL-CONF-192528.
26. Yoshida, Y., Suzuki, J., Hirao, T., & Nagata, M. Dependency-based Discourse Parser for Single-Document Summarization //EMNLP. – 2014. – C. 1834-1839.
27. Carenini G., Cheung J. C. K., Pauls A. MULTI-DOCUMENT SUMMARIZATION OF EVALUATIVE TEXT //Computational Intelligence. – 2013. – T. 29. – №. 4. – C. 545-576.
28. Gerani, S., Mehdad, Y., Carenini, G., Ng, R. T., & Nejat, B. Abstractive Summarization of Product Reviews Using Discourse Structure //EMNLP. – 2014. – C. 1602-1613.
29. Zirn, C., Niepert, M., Stuckenschmidt, H., & Strube, M. Fine-Grained Sentiment Analysis with Structural Features //IJCNLP. – 2011. – C. 336-344.
30. Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., & de Jong, F. Polarity analysis of texts using discourse structure //Proceedings of the 20th ACM international conference on Information and knowledge management. – ACM, 2011. – C. 1061-1070.
31. Feng, V. W., Lin, Z., Hirst, G., & Holdings, S. P. The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence //COLING. – 2014. – C. 940-949.
32. Feng V. W. RST-style discourse parsing and its applications in discourse analysis :дис. – University of Toronto, 2015.
33. Feng V. W., Hirst G. Authorship Verification with Entity Coherence and Other Rich Linguistic Features.
34. Feng V. W., Hirst G. Patterns of local discourse coherence as a feature for authorship attribution //Literary and Linguistic Computing. – 2014. – T. 29. – №. 2. – C. 191-198.
35. Joyce Y. Chai and RongJin. Discourse structure for context question answering. In SandaHarabagiu and Finley Lacatusu, editors, HLT-NAACL 2004: Workshop on Pragmatics

of Question Answering, pages 23–30, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

36. Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter Arno Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 735–736. ACM.

37. Galitsky, B., Ilvovsky, D., Kuznetsov, S., & Strok, F. Matching sets of parse trees for answering multi-sentence questions // Proc. Recent Advances in Natural Language Processing (RANLP 2013), Bulgaria. – 2013.

38. Кибрик, А. А., Добров, Г. Б., Залманов, Д. А., Линник, А. С., & Лукашевич, Н. В. Референциальный выбор как многофакторный вероятностный процесс // По материалам международной конференции «Диалог». – 2010. – С. 173-180.

39. Литвиненко А. О. Описание структуры дискурса в рамках Теории Риторической Структуры: применение на русском материале // Труды Международного семинара Диалог. – 2001. – С. 159-168.

ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ КАТЕГОРИЙ «РЕСУРС», «СВОЙСТВО», «ДЕЙСТВИЕ», «ОТНОШЕНИЕ»

Колесников А.В.

БФУ им. И.Канта, Калининградский филиал ФИЦ ИУ РАН
Калининград, ул. Невского, д. 14, avkolesnikov@yandex.ru

Листопад С.В.

Калининградский филиал ФИЦ ИУ РАН, БФУ им. И.Канта
Калининград, ул. Гостиная, д. 5, ser-list-post@yandex.ru

Денисова А.С.

БФУ им. И.Канта

Калининград, ул. Невского, д. 14, annadenisova312@mail.ru

Колодин А.Е.

БФУ им. И.Канта

Калининград, ул. Невского, д. 14, alex_kolodin@mail.ru

Аннотация

Гибридные интеллектуальные системы, то есть системы, в которых для решения задачи используется более одного метода имитации интеллектуальной деятельности человека, за счет моделирования знаний нескольких экспертов и процессов их взаимодействия позволяют решать задачи с высокой сложностью моделирования. Тем не менее, элементы современных гибридных интеллектуальных систем взаимодействуют на основе символично-логических моделей знаний, что существенно ограничивает их возможности по сравнению с коллективами экспертов, успешно оперирующих как символично-логическими, так и визуально-образными знаниями. Моделирование коллективных рассуждений на основе визуально-образной информации предлагается реализовать в новом классе гибридных интеллектуальных систем - гибридных интеллектуальных систем с гетерогенным визуальным полем. Создание таких систем предполагает разработку аксиоматической теории ролевых визуальных

моделей объекта, субъекта визуального управления и семейства ролевых визуальных моделей на категориальном ядре «ресурс – свойство – действие - отношение». Настоящая работа посвящена вопросам визуального моделирования категориального ядра.

1. Введение

Визуальные языки – важная часть автоматизированных систем разработки программного обеспечения, моделирования бизнес-процессов, поддержки принятия решений и других. Такие языки позволяют наглядно описать структуру, свойства и отношения понятий предметной области. Основное различие естественных и визуальных языков, обусловившее популярность последних при построении моделей предметных областей, – размерность «пространства» языка. В то время как, предложения естественного языка по своей природе одномерны, рисунки, схемы и диаграммы визуальных языков имеют не менее двух измерений [1]. Различие в размерностях языков обуславливает различия в выразительных возможностях и степени сложности восприятия человеком представленной информации. Известна поговорка «лучше один раз увидеть, чем сто раз услышать». Правильно составленный график или диаграмму значительно проще анализировать, чем многостраничную таблицу с результатами измерений, а тем более текстовое их описание.

Другая особенность естественных языков профессиональной деятельности состоит в том, что они предназначены в основном для реализации символично-логических рассуждений, т.е. рассуждений в рамках абстрактных цепочек символов, с которыми связаны некоторые семантические и прагматические представления. Однако, как показывают исследования [2], не менее важен и другой механизм мышления – способность использовать визуальные образы и представления о них. Такие образы обладают большей конкретностью и интегрированностью, чем символические представления. Они и значительно более расплывчаты, менее логичны, чем то, что скрывается за элементами, с которыми оперирует логическое мышление. Без них нельзя отображать в сознании человека окружающий мир в той полноте, которая для него характерна. Именно визуальные языки предназначены для реализации образных рассуждений.

В области обработки визуальных образов известны работы Д.А. Поспелова, Г.П. Щедровицкого, Ю.Р. Валькмана, Б.А. Кобринского, О.П. Кузнецова, Г.С. Осипова, В.Б. Тарасова, И.Б. Фоминых, Т.А. Гавриловой, А.Е. Янковской. В настоящее время, основные исследования в области визуальных языков сосредоточены на разработке специальных визуальных языков для представления определенных видов данных, информации и знаний. Визуальные языки были разработаны для функционального программирования, программирования на примерах, конечных автоматов, потоков данных и других областей [3]. Реализация этих языков требует разработки для каждого случая специальных сред создания, манипулирования и обработки визуальных моделей, что требует значительных усилий. Одна из причин такой ситуации – отсутствие формализованных моделей для определения визуального синтаксиса и инструментов языка. В этой связи предлагается разработать визуальный язык на основе принципов теории систем и системного анализа.

2. Неформальная аксиоматическая теория

Практика работы с языками профессиональной деятельности (ЯПД) в технических, социально-экономических и био-производственных системах [4 – 6] показывает, что основные проблемы их анализа — неструктурированность, несистематизирован-

ность знаний, неоднородность, противоречивость терминологии, специальная лексика. На ЯПД построен документооборот в системе управления, сформулированы знания, зафиксированные в учебниках, монографиях, методиках, стандартах, инструкциях, рекламных проспектах, каталогах услуг и т.п., являющиеся одним из ценных вербализованных источников информации при моделировании. Таким образом, задача, выраженная на ЯПД, и машинная программа — два полюса шкалы «оригинал-программа». На этой шкале располагаются и модели, занимающие промежуточное положение. Актуальность «промежуточных» моделей объясняется невозможностью в настоящее время преобразовать информацию, записанную на ЯПД, в информацию, «понятную» ЭВМ. Поэтому современная методология системного анализа предусматривает последовательное и, зачастую, итерационное моделирование на искусственных языках, свободных от недостатков ЯПД. Один из таких искусственных языков построения моделей, расположенных по шкале «оригинал-программа» ближе к левой границе, — язык концептуальных моделей.

Концептуальная модель (англ. concept — понятие, идея; общее представление; концепция, лат. conceptio — понимание, система) — модель предметной области, состоящая из: 1) перечня всех понятий, используемых для описания этой области, вместе со свойствами и характеристиками, классификацией этих понятий по типам, ситуациям, признакам в данной области и 2) законов протекания процессов в ней [7].

В основу концептуализации было положено описание предметной области в виде неформальной (содержательной) аксиоматической теории Te , включающей четыре множества [8]:

$$Te = \langle Te^{TH}, Te^{TO}, Te^A, Te^{BT} \rangle, \quad (1)$$

где Te^{TH} , Te^{TO} — неопределяемые и определяемые термины соответственно; Te^A — аксиомы; Te^{BT} — высказывания-теоремы, выводимые из Te^A по некоторым фиксированным логическим правилам. Для построения теории (1) зададим, прежде всего, Te^{TH} , Te^{TO} , т.е. ее категориальное ядро.

Один из результатов в теории систем и системном анализе получен А.И. Уёмовым [9], предложившим в качестве базиса системного подхода и языка анализа и синтеза сложных объектов триаду категорий «вещь — свойство — отношение». В своих работах А.И. Уёмов для раскрытия структуры окружающего мира главную роль отводит категориям вещи, свойства и отношения. Главная особенность триады «вещь — свойство — отношение» состоит в соотносительности, т.е. в том, что все эти категории определяются друг через друга, причем центральная, основная категория — категория вещи [9]. Из соотносительности следует, что в категориальном ядре не действует правило о запрещении круга в определениях и что в неформальной аксиоматической теории (1) множество $Te^{TH} = Te^{TO} = \{\text{«вещь»}, \text{«свойство»}, \text{«отношение»}\}$.

Дадим следующие определения категориям ядра:

1. Вещь — это система качеств. Понимая вещь качественно, будем предполагать следующее: различные вещи — это различные системы качеств, одна и та же вещь — это одна и та же система, качеств, все вещи существуют в пространстве и времени. Например, кран, бригада, судно, почва, рефрижератор, товар.

2. Свойство — все то, что не является границами данной вещи. Это то, что, характеризуя вещи, не образует новых вещей. Существенное свойство вещи, граница вещи — качество. С исчезновением качества одна вещь превращается в другую. Одно и то же качество может быть присуще разным вещам. Одна и та вещь может обладать множе-

ством различных качеств, каждое из которых отделяет ее от других вещей. Например, свойства – производительность, грузоподъемность, мощность, а качества – проект судна, конструкция крана, возможность охлаждения груза.

3. Отношение — то, что образует вещь из данных элементов (свойств или других вещей). Отношение — есть то, что, будучи установлено между вещами, образует новые вещи. Например, равенство, сходство, рядом, после.

Однако непосредственное применение данного категориального ядра Te^{TH} к системному анализу предметной области затруднительно в силу нескольких причин. Во-первых, категория вещи — верхний предел абстракции, что вызывает трудности с ее интерпретацией разработчиком. Во-вторых, имеются трудности с интерпретацией понятия «зависимость» между категориями ядра. Поскольку зависимость между ними — это отношение, то возникает неоднозначность в толковании отношений. Если использовать язык дискретной математики, то многозначность проявляется в том, что отношения обозначаются вершиной графа-треугольника и его ребрами одновременно. В-третьих, категория «отношение» носит общий характер и целесообразно выбрать категорию, которая, являясь отношением, была бы в то же время предметно-ориентированной.

Трудности преодолеем следующим образом. Из множества вещей будем рассматривать только вещи-ресурсы (в дальнейшем просто ресурсы), подчеркивая тем самым, что прагматичны не вещи сами по себе, а только вещи, имеющиеся в распоряжении для решения задач. Будем также считать, что природные явления также ресурс, вне зависимости от положительности или отрицательности эффекта их воздействия.

Из множества отношений выберем отношения-действия (далее — просто действия), подчеркивая тем самым интерес к отношениям на ресурсах, установление или разрыв которых может изменить состояние объекта управления, т.е. к операциям и процессам, в частности технологическим и производственным.

Таким образом, возможны следующие определения. Ресурс — понятие, обозначающее вещь, имеющуюся у субъекта управления для решения задач. Действие — понятие, обозначающее отношения на ресурсах, как следствие деятельности, поступков и поведения.

Введем множество $X = \{RES, PR, ACT\}$, где $RES = X^{res}$, $PR = X^{pr}$, $ACT = X^{act}$ — множества базисных понятий, обозначающих ресурсы, свойства и действия, соответственно, и построим на X полный граф $G = \langle X, R \rangle$. Тогда имеем классификацию базовых отношений $R = \{R^{res\ res}, R^{pr\ pr}, R^{act\ act}, R^{res\ pr}, R^{pr\ res}, R^{res\ act}, R^{act\ res}, R^{act\ pr}, R^{pr\ act}\}$, где $R^{res\ res}$, $R^{pr\ pr}$, $R^{act\ act}$, $R^{res\ pr}$, $R^{pr\ res}$, $R^{res\ act}$, $R^{act\ res}$, $R^{act\ pr}$, $R^{pr\ act}$ — множества отношений «ресурс-ресурс», «свойство-свойство», «действие-действие», «ресурс-свойство», «свойство-ресурс», «ресурс-действие», «действие-ресурс», «действие-свойство», «свойство-действие» соответственно.

Таким образом, после того как определен граф G , имеем $Te^{TH} = \{\text{«вещь»}, \text{«свойство»}, \text{«отношение»}\}$, $Te^{TO} = \{\text{«вещь»}, \text{«свойство»}, \text{«отношение»}, \text{«ресурс»}, \text{«действие»}\}$ и можно ввести аксиоматику теории (1):

Аксиома 1 (аксиома существования одноролевого конструкта). Если $X^\alpha \subseteq X$, а $R^{\alpha\ \alpha} \subseteq R$, то существует одноролевой конструкт co^1 , которым назовем выражение $co^1 = R^{\alpha\ \alpha}(X^\alpha, X^\alpha)$, где $\alpha \in \{\text{«res»}, \text{«pr»}, \text{«act»}\}$. Одноролевой конструкт имеет в качестве ролей одну и ту же категорию.

Аксиома 2 (аксиома существования двухролевого конструкта). Если $X^\alpha, X^\beta \subseteq X$, а $R^{\alpha\alpha}, R^{\beta\beta}, R^{\alpha\beta} \subseteq R$, то существует двухролевой конструкт co^2 , которым назовем выражение $co^2 = R^{\alpha\alpha}(X^\alpha, X^\alpha) \circ R^{\beta\beta}(X^\beta, X^\beta) \circ R^{\alpha\beta}(X^\alpha, X^\beta)$, где $\alpha, \beta \in \{ "res", "pr", "act" \}$, $\alpha \neq \beta$, знак \circ обозначает конкатенацию.

Аксиома 3 (аксиома существования трехролевого конструкта). Если $X^\alpha, X^\beta, X^\gamma \subseteq X$, а $R^{\alpha\alpha}, R^{\beta\beta}, R^{\gamma\gamma}, R^{\alpha\gamma}, R^{\beta\alpha}, R^{\gamma\beta} \subseteq R$, то существует трехролевой конструкт co^3 , которым назовем выражение $co^3 = R^{\alpha\alpha}(X^\alpha, X^\alpha) \circ R^{\beta\beta}(X^\beta, X^\beta) \circ R^{\gamma\gamma}(X^\gamma, X^\gamma) \circ R^{\alpha\gamma}(X^\alpha, X^\gamma) \circ R^{\beta\alpha}(X^\beta, X^\alpha) \circ R^{\gamma\beta}(X^\gamma, X^\beta)$, где $\alpha, \beta, \gamma \in \{ "res", "pr", "act" \}$, $\alpha \neq \beta \neq \gamma$.

В итоге, имеем $Te^A = \{A_1, A_2, A_3\}$ и три множества конструктов $CO^1 = \{co_1^1, \dots, co_3^1\}$, $CO^2 = \{co_1^2, \dots, co_6^2\}$, $CO^3 = \{co_1^3, \dots, co_8^3\}$. На категориальном ядре, т.е. множестве X , невозможно задать никаких других конструктов.

Для моделирования предметных знаний об объекте управления в [6] категориальное ядро расширяется дополнительными, производными категориями $PAR = X^{par} \subseteq X^{pr}$ — физическими свойствами (параметрами), $CH = X^{ch} \subseteq X^{pr}$ — характеристическими свойствами (характеристиками), $NAM = X^{nam} \subseteq X^{pr}$ — именными свойствами (именами), $MES = X^{mes}$ — единицами измерения (мерами), $VAL = X^{val}$ — значениями и $ST = X^{st}$ — состояниями. Для моделирования предметных знаний о субъекте управления в X включены $EST = X^{est}$ — оценки и $PRB = X^{prb}$ — задачи, а о субъекте моделирования — $MET = X^{met}$ — метод и $MOD = X^{mod}$ — модель. Кроме того, чтобы иметь возможность расширять категориальное ядро, введем $EX = X^{ex}$ — экзотические понятия.

Представленная неформальная аксиоматическая теория позволяет строить формализованные модели предметной области, избавляясь от таких недостатков ЯПД как, неструктурированность и несистематизированность знаний. Однако в результате устранения одних недостатков возникли другие: сложность построения и сопровождения концептуальных моделей предметной области из-за их ненаглядности и громоздкости выражений, необходимости ведения «словаря» концептов. С целью устранения этих недостатков предлагается графический язык представления концептуальных моделей, основанный на неформальной аксиоматической теории.

3. Графическое представление ролевых концептуальных моделей

Визуализация – мощный инструмент, предназначенный для организации и облегчения процесса познания [10]. Визуальные модели обладают особой познавательной силой, фактически представляя собой средства когнитивной графики для структурирования информации. На практике, используются более сотни методов визуального структурирования, что обусловлено существенными различиями в природе, особенностях и свойствах знаний различных предметных областей.

В [11] представлен интересный подход к классификации методов визуализации с использованием своеобразной «таблицы Менделеева». Несмотря на то, что расположение методов в таблице весьма условное и, очевидно, не может претендовать на строгость периодической системы элементов Д.И. Менделеева, авторам удалось наглядно

классифицировать 100 методов визуализации информации по нескольким важным признакам: 1) область применения метода: визуализация данных, информации, концептов, стратегий, метафор и смешанная; 2) вид информации, представляемой методом: структура или процесс; 3) степень обобщенности представляемой информации: обзорная, детальная или смешанная; 4) вид порождаемых когнитивных процессов: дивергентный или конвергентный.

Визуализация модели предметной области может выполняться методами семантических сетей, интеллектуальных карт, концептуальных вееров, концептуальных карт и другими [11]. Традиционно для описания концептуальных моделей предметной области применяются концептуальные карты (диаграммы). Такие диаграммы отображают концепты, идеи или информацию в виде прямоугольников или окружностей, которые связаны маркированными стрелками, представляющими отношения между ними. Отношения между концептами обычно формулируются с использованием связывающих фраз, например, «вызывает», «обуславливает», «требует», «способствует» и других.

Концептуальные карты были предложены Дж.Д. Новаком в 1970-х гг. [12] для оценки понимания предметных знаний школьников. Во многом они напоминают семантические сети, отличаясь от них в основном способом представления элементов. Также концептуальные карты похожи на интеллектуальные карты, однако, последние предназначены в основном для представления радиальных иерархий и древовидных структур.

В данной работе, учитывая ролевой характер используемых концептуальных моделей, для графического представления предлагается использовать модифицированные концептуальные карты, отображающие не только наименование концепта, но и его роль, категорию. В такой диаграмме понятия категорий «ресурс», «действие» и «свойство» изображаются в овальном блоке, как показано на рис. 1, а. В верхней части блока указывается категория, а в нижней части — само понятие. Отношения между понятиями категорий «ресурс», «действие» и «свойство» представляются в виде стрелки (рис. 1, б), рядом с которой указывается наименование отношения.

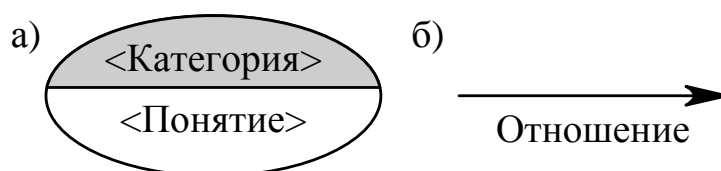


Рис. 1. – Условные обозначения, используемые в концептуальных моделях:
 а) блок, отображающий понятия категорий «ресурс», «действие» и «свойство»;
 б) стрелка, отображающая отношение

Практика построения графических концептуальных моделей показала, что при описании предметной области на графах часто встречаются однотипные фрагменты. В качестве примера можно привести следующий фрагмент, записанный в общем виде, устанавливающий связь между свойством некоторого концепта и классом переменной (табл. 1.3 в [6]), с помощью которой оно представляется:

$$r_1^{pr\ pr}(pr_j, pr_3) \circ r_1^{pr\ val}(pr_3, val_k), \quad (2)$$

где $pr_j \in PR$ – произвольный концепт категории «свойство», $pr_3 \in PR$ – концепт «класс» категории «свойство», $val_k \in VAL$ – концепт «имя класса» категории «значение», $r_1^{pr\ pr}$ – отношение «иметь свойство» категории «свойство – свойство», $r_1^{pr\ val}$ – отношение «иметь значение» категории «свойство – значение». Такие фрагменты концептуальной модели будем называть подграфами «свойство – класс». Очевидно, что с усложнением модели и ростом числа свойств каждого из представленных в модели концептов число таких подграфов будет увеличиваться и загромождать модель. Для того чтобы избежать этого, введем новый тип концептов (вершин), с помощью которых будем представлять подграфы «свойство — класс» и которые назовем концептами (вершинами) «свойство — класс».

Множество концептов «свойство — класс» (2) обозначим $PT = \{pt_{j\ k} \mid j=1, \dots, N_{pr}, j \neq 3, k=1, \dots, N_{val}\}$, где N_{pr} – число концептов категории «свойство» в модели, N_{val} – число концептов категории «значение» в модели. Произвольный концепт «свойство — класс» $r_1^{pr\ pr}(pr_j, pr_3) \circ r_1^{pr\ val}(pr_3, val_k)$, где $pr_j \in PR, val_k \in VAL$, записывается: $pt_{j\ k} = (pr_j, pr_3, val_k)$. Множество вершин подграфа «свойство — класс» $pt_{j\ k}$ обозначим $VPT_{j\ k} = \{pr_j, pr_3, val_k\}$, а множество отношений — $EPT_{j\ k} = \{r_1^{pr\ pr}(pr_j, pr_3), r_1^{pr\ val}(pr_3, val_k)\}$.

Графически замена подграфа «свойство — класс» вершиной «свойство — класс» представлена на рис. 2.

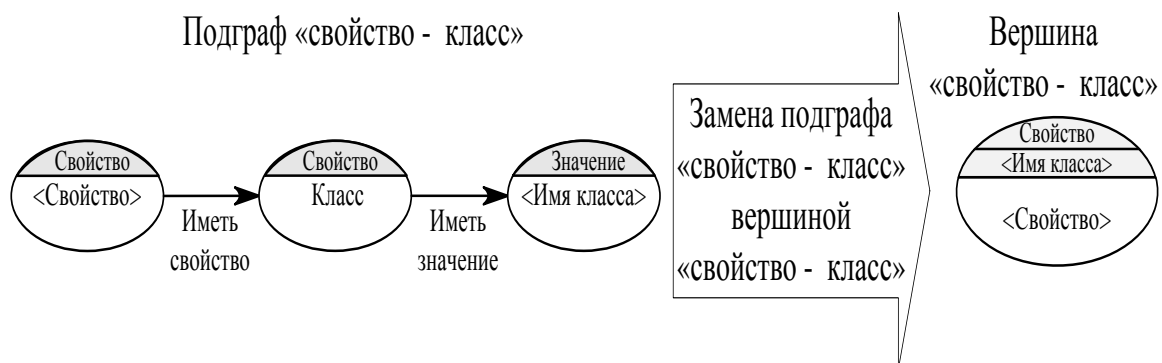


Рис. 2. – Замена подграфа «свойство — класс» вершиной «свойство — класс»

Аналогичным образом может быть выполнено представление типовых подграфов, в случае их выявления, соответствующими вершинами.

Нужно отметить, что при построении реальных моделей предметных областей с использованием концептуальных карт получаемые диаграммы могут быть достаточно громоздкими и нечитаемыми. Это сводит на нет все преимущества графического представления модели по сравнению текстовым. В этом случае, модель предметной области следует рассматривать на различных уровнях обобщенности, разрабатывая для каждого уровня свою концептуальную карту и организовывая эти карты в иерархию с помощью одного или нескольких методов визуального представления информации, например, с помощью многослойных диаграмм [11].

4. Результаты практического применения неформальной аксиоматической теории ролевых концептуальных моделей и дальнейшие перспективы

Построение концептуальных моделей в рамках неформальной аксиоматической теории выполняется с использованием схем ролевых концептуальных моделей [5, 6]. Схемы ролевых концептуальных моделей строятся из конструкторов-аксиом, и есть высказывания-теоремы неформальной аксиоматической теории, полученные по правилу склеивания [5, 6]. Схемы ролевых концептуальных моделей — эвристики, которые могут использоваться с разными целями:

1) для построения в памяти компьютера семантической структуры текстов [13] на ЯПД экспертов, ЛПР и разработчиков, для разработки модели действительности [14] с целью автоматизированного преобразования схемы ролевых концептуальных моделей в ролевые концептуальные модели;

2) для структурированного представления знаний о сложных задачах;

3) для структурированного представления знаний о методах и моделях;

4) для структурированного представления знаний об элементах гибридных интеллектуальных систем (ГиИС), а также ГиИС в целом;

5) для структурированного представления знаний о содержании и логической последовательности действий разработчика ГиИС для описания системы операций, разработки информационной и инструментальной сред технологии функциональных ГиИС.

Чтобы преобразовать схему ролевой концептуальной модели в концептуальную ролевую модель, необходимо наполнить классы понятий и отношений из базового и расширенного ядра теории Te (1) лексемами и специфицировать элементы множеств X, R , использованных в схеме, т.е. выполнить интерпретацию теории Te (1).

В качестве примера ролевых концептуальных моделей, построенных в рамках неформальной аксиоматической теории можно привести следующие: модели классической (рис. 3) и неклассических задач коммивояжера [6], модель сложной транспортно-логистической задачи [15], модель сложной задачи оперативно-производственного планирования [16], модель задачи сложной задачи диагностики артериальной гипертензии [17], модели ГиИС и ее элементов [5], модели гибридной интеллектуальной многоагентной системы и ее агентов [15], модели виртуального гетерогенного коллектива [18] и другие.

Применение ролевых концептуальных моделей для описания предметной области позволило систематизировать и структурировать понятия естественного ЯПД и отношения между ними. Разработанные ролевые концептуальные модели легли в основу онтологических моделей предметных областей информационных систем. За счет того что все элементы таких информационных систем используют общую модель предметной области, существенно упрощается процесс их сопровождения и модификации: корректировка модели предметной области, добавление в нее новых знаний отражается на поведении всех элементов системы.

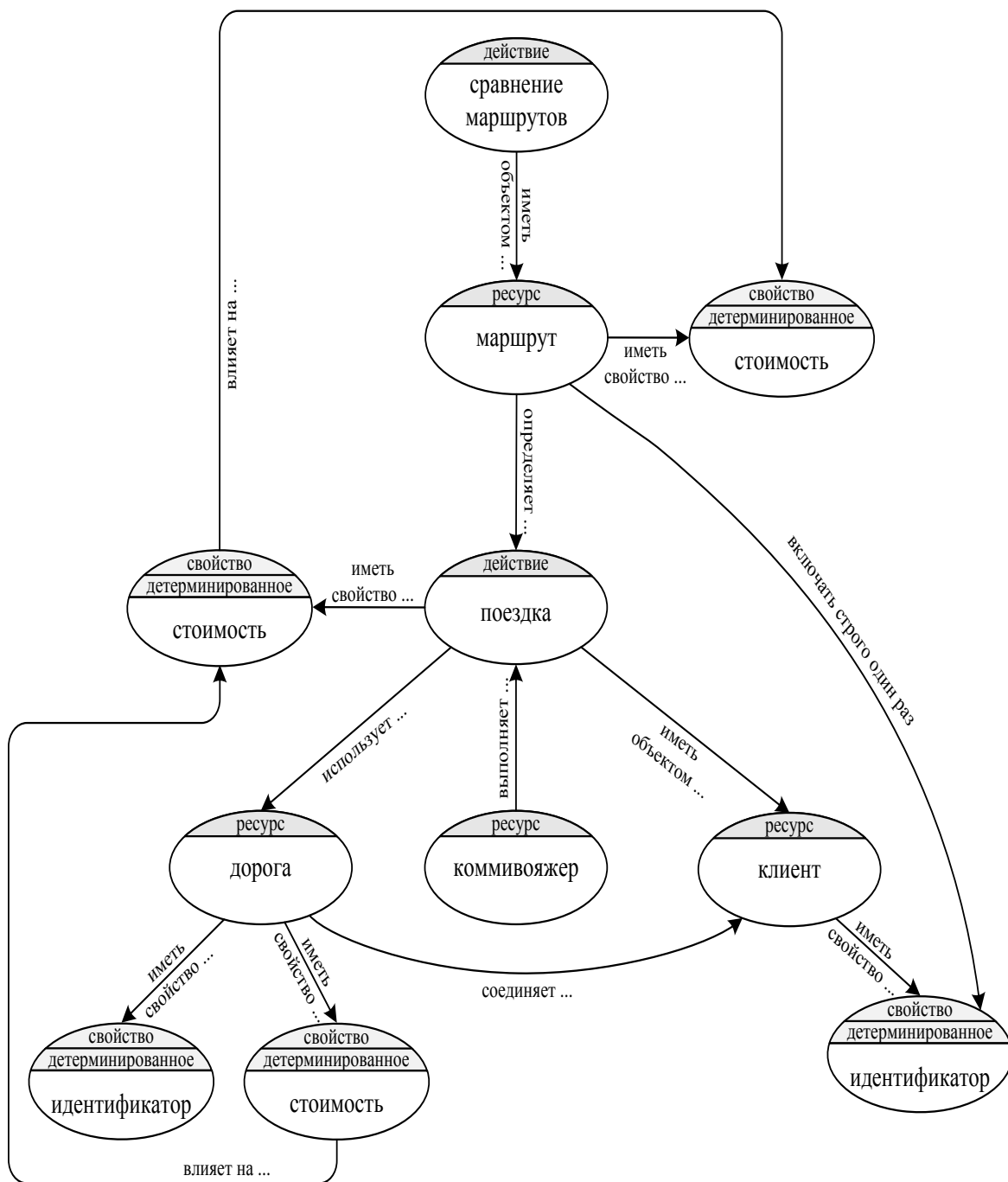


Рис. 3. – Граф концептуальной модели решения классической задачи коммивояжера

Применение ролевых концептуальных моделей для моделирования методов решения задач за счет формализованного описания элементов этих методов и отношений между ними позволило структурировать, организовать деятельность разработчиков этих методов. Применение этих моделей привело к появлению таких методов решения сложных задач как гибридная интеллектуальная система с координацией элементов, гибридная интеллектуальная многоагентная система и другие.

Использование графического представления ролевых концептуальных моделей позволило значительно упростить и ускорить процесс построения таких моделей. В частности, опыт разработки ролевых концептуальных моделей [5, 6, 15 – 18] показывает, что использование их графической интерпретации ускоряет процесс разработки модели в 3 – 5 по сравнению с символьным представлением. При этом графическое представление ролевой концептуальной модели имеет однозначную символьную интерпретацию, оба вида концептуальных моделей взаимозаменяемы. Это позволяет визуализировать уже разработанные символьные модели и строить формализованные символьные модели на основе графических, применяя разработанные для символьных моделей методы построения онтологий интеллектуальных информационных систем.

Визуализация категориального ядра «ресурс – свойство – действие – отношение» будет положена в основу нового класса функциональных гибридных интеллектуальных систем, имитирующих работу коллективного интеллекта по поиску решений над гетерогенными модельным и визуальным полями. При его разработке будут применяться знания и опыт, извлеченные из работ по визуальному управлению, когнитивной графике, инженерии образов, изобразительному искусству и искусству составления презентаций.

Предлагаемый класс функциональных гибридных интеллектуальных систем с гетерогенным визуальным полем, как искусственный коллективный интеллект, будет иметь следующие преимущества:

1) управление имитационным процессом в зависимости от неопределенности проблемной ситуации: когда область явлений формализована (частично формализована) подключать для поиска решений знания экспертов из гетерогенного модельного поля; когда есть существенная неопределенность не снимаемая точным анализом и логико-математическими рассуждениями, привести в действие механизмы визуально-пространственного, образного мышления, что позволит имитировать «скачки» в гибридном пространстве состояний функциональной гибридной интеллектуальной системы, соответствующие мгновенному интуитивному инсайту, озарению, прерывающему логико-математические рассуждения;

2) анализ характеристик объекта визуального управления, как в совокупности, так и в отдельности за счет комбинирования логико-математических и визуально-пространственных представлений, что даст более полную картину наблюдения за объектом;

3) наблюдение за поведением отдельного виртуального индивида в контексте наблюдения за поведением искусственного коллективного интеллекта, поможет получить новые знания о феномене естественного коллективного интеллекта, обнаружить и предотвратить ошибки в решениях экспертов, корректируя тем самым общее решение коллектива;

4) существенное снижение нагрузки на операторов визуального управления, поскольку визуально-пространственное мышление отображает в сознании окружающий мир в той полноте, которая характерна для человека, когда достаточно одного взгляда для понимания условий возникновения проблемной ситуации в объекте управления и оценки степени риска сохранения ненормативного поведения.

5. Заключение

В работе представлена неформальная (содержательная) аксиоматическая теория ролевых концептуальных моделей, в которой выполнено моделирование логико-математического коллективного искусственного интеллекта, что составило основу лингвистического подхода к гибридизации. Применение этой теории позволяет избавиться при построении ролевых концептуальных модели предметной области от таких недостатков ЯПД как, неструктурированность и несистематизированность знаний.

Представленный подход к визуализации категориального ядра «ресурс – свойство – действие – отношение» позволил значительно упростить и ускорить процесс построения концептуальных моделей на его основе. Визуализация категориального ядра «ресурс – свойство – действие – отношение» будет положена в основу нового класса функциональных гибридных интеллектуальных систем с гетерогенным визуальным полем, что позволит получить новые теоретическое, методологическое и технологические знания о коллективном интеллекте по инфокоммуникационным технологиям визуального управления.

Работа поддержана Российским фондом фундаментальных исследований (проект 16-07-00271а).

Литература

1. Tveit M.S. Specification of Graphical Representations - using hypergraphs or meta-models? <<http://www.nik.no/2008/04-Tveit.pdf>>. 18.03.2016.
2. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. — 304 с.
3. Golin E.J., Reiss S.P. The Specification of Visual Language Syntax // Journal of Visual Languages and Computing, 1990, Vol. 1, pp. 141-157.
4. Колесников А.В. Гибридные интеллектуальные системы. Теория и технология разработки. – СПб.: СПбГТУ, 2001. – 711 с.
5. Колесников А.В., Кириков И.А. Методология и технология решения сложных задач методами функциональных гибридных интеллектуальных систем. – М.: ИПИ РАН, 2007. – 387 с.
6. Колесников А.В., Кириков И.А., Листопад С.В., Румовская С.Б., Доманицкий А.А. Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем. – М.: ИПИ РАН, 2011. – 295 с..
7. Толковый словарь по искусственному интеллекту / А.Н. Аверкин, М.Г. Гаазе–Рапопорт, Д.А. Поспелов. — М.: Радио и связь, 1992. — 256 с.
8. Лелюк В.А. Концептуальное проектирование систем с базами знаний. – Харьков: Основа, 1990. – 144 с.
9. Уёмов А.И. Вещи, свойства, отношения. – М.: Институт философии АН СССР, 1963. – 184 с.
10. Гаврилова Т.А., Гулякина Н.А. Визуальные методы работы со знаниями: попытка обзора // Искусственный интеллект и принятие решений, 2008, № 1, С. 15 – 21.
11. Lengler R., Eppler M. A Periodic Table of Visualization Methods. <http://www.visual-literacy.org/periodic_table/periodic_table.html>. 18.03.2016.
12. Novak J.D. A theory of education. – Ithaca, NY: Cornell University Press, 1977.
13. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.

14. Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. – М.: Наука, 1997. – 112 с.
15. Колесников А.В., Кириков И.А., Листопад С.В. Гибридные интеллектуальные системы с самоорганизацией: координация, согласованность, спор. – М.: ИПИ РАН, 2014. – 189 с.
16. Колесников А.В., Солдатов С.А. Теоретические основы решения сложной задачи оперативно-производственного планирования с учетом координации // Вестник Российского государственного университета им. И. Канта, 2009, Вып. 10, С. 82 – 96.
17. Кириков И.А., Колесников А.В., Румовская С.Б. Исследование сложной задачи диагностики артериальной гипертензии в методологии искусственных гетерогенных систем // Научный журнал «Системы и средства информатики», 2013, Т. 23, Вып. 2, С. 96 – 114.
18. Кириков И.А., Колесников А.В., Румовская С.Б. Функциональная гибридная интеллектуальная система для поддержки принятия решений при диагностике артериальной гипертензии // Системы и средства информатики, 2014, Т. 24, Вып. 1, С. 153 – 179.

МЕТОД ОБРАБОТКИ ПАРНЫХ СРАВНЕНИЙ ВАРИАНТОВ, ОПИСАННЫХ МУЛЬТИМНОЖЕСТВАМИ

Малтугеева Галина Станисавовна

Институт динамики систем и теории управления им.В.М.Матросова СО РАН

г. Иркутск, ул. Лермонтова, 134

gama@icc.ru

Петровский Алексей Борисович

Федеральный исследовательский центр «Информатика и управление» Российской

академии наук Институт системного анализа

г. Москва, проспект 60-летия Октября, 9.

pab@isa.ru

Аннотация

В работе исследован вопрос обработки матриц парных сравнений, в которых оценки могут иметь как числовые, так и вербальные значения. Предложен способ обработки информации такого вида, основанный на применении аппарата мультимножеств. Данный способ позволит одновременно обрабатывать как числовые, так и вербальные значения в матрицах парных сравнений. Способ может быть включен в состав многометодного подхода к агрегированию индивидуальных предпочтений.

1. Введение

Задача принятия решений является одной из часто решаемых как в повседневной жизни, так и в профессиональной деятельности. Для обеспечения поддержки лицу, принимающему решение, разработаны многочисленные методы выбора, методологии принятия решения, а также разнообразное программное обеспечение. Для применения того или иного инструментария необходимо определить тип решаемой задачи, форму представления входных данных и результата.

В общем виде задача принятия решения представляет совокупность множества вариантов решения, список признаков (параметров, атрибутов) вариантов, индивидуальные предпочтения одного или нескольких экспертов. Трудоемкость процесса решения задачи возрастает при увеличении числа рассматриваемых признаков и в случае необходимости обобщения мнений группы экспертов. В литературе описаны разнообразные способы решения задач индивидуального и группового выбора, однокритериального и многокритериального принятия решений, каждый из которых имеет особенности (форма индивидуальных предпочтений, форма результата решения, требования к экспертам и лицу, принимающему решение).

Наиболее интересной для исследований является задача группового многокритериального выбора, в которой требуется агрегировать мнения группы экспертов относительно набора признаков вариантов. Индивидуальные предпочтения экспертов могут быть представлены в виде оценок (вербальных, числовых), упорядоченных вариантов, парных сравнений. Для решения задач такого вида разработаны следующие методы [1]: метод усреднения индивидуальных оценок, метод аддитивной свертки индивидуальных ценностей, метод групповой аналитической иерархии, метод агрегирования парных сравнений (РАМПА), метод

оценки близости к опорной точке с усредненными оценками (TOPSIS), метод ценки близости к опорной точке с суммарными оценками (АРАМИС), метод агрегирования индивидуальных ранжировок (АИР). Большинство этих методов предназначены для обработки оценок вариантов, при этом два из них (АРАМИС, АИР) базируются на использовании мультимножеств в метрическом пространстве Петровского [2].

По мнению психологов, наибольшего доверия как форма представления индивидуальных предпочтений человека заслуживают парные сравнения вариантов. При чем в процессе парных сравнений могут использоваться как числовые, так вербальные значения. В литературе широко описаны методы обработки матриц парных сравнений с числовыми оценками. Предлагается разработать метод обработки матриц парных сравнений с применением аппарата мультимножеств, который позволит обрабатывать парные сравнения как с числовыми, так и с вербальными оценками.

2. Постановка задачи

В рассматриваемой задаче группового многокритериального принятия решений известно, что:

- множество вариантов $A = \{A_1, A_2, \dots, A_n\}$ задано априори, конечно и не изменяется в процессе решения задачи;
- перечень критериев $C = \{C_1, C_2, \dots, C_m\}$, характеризующих признаки вариантов;
- группа экспертов $Expert = \{Expert_1, Expert_2, \dots, Expert_l\}$, осуществляющих попарные сравнения вариантов по каждому критерию;
- определены шкалы оценивания X_1, X_2, \dots, X_m для попарных сравнений по каждому критерию, имеющие числовые или вербальные градации значений;

Требуется построить агрегированные оценки вариантов, которые позволят сделать окончательный выбор лицу, принимающему решение.

Результат парных сравнений вариантов по каждому критерию предствляется в виде матрицы парных сравнений, общий вид которой представлен в таблице 1, где a_{ij} – может иметь как числовое, так и вербальное значение.

Таблица 1 – Общий вид матрицы парных сравнений

	A_1	A_2	...	A_n
A_1	–	a_{12}		a_{1n}
A_2	a_{21}	–		a_{2n}
...				
A_n	a_{n1}	a_{n2}		–

Число a_{ij} может означать [3]:

1. наличие факта предпочтительности (непредпочтительности) варианта A_i по сравнению с A_j ;
2. балльную оценку предпочтительности варианта A_i над A_j ;
3. долю суммарной интенсивности предпочтения двух этих объектов, приходящуюся на A_i ;
4. по сколько раз A_i предпочтительнее A_j .

Следует отметить, что все числовые значения могут быть приведены к четвертому типу. Предполагается, что все оценки в парных сравнениях должны быть связаны некоторыми соотношениями.

В литературе достаточно широко описаны методы обработки и их применение к решению практических задач с использованием числовых значений в матрицах парных сравнений. Все они сводятся к применению допустимых преобразований матриц, содержащих исключительно числовые показатели предпочтительности объектов. Описание методов обработки парных сравнений вариантов, имеющих вербальные значения, отсутствует.

3. Мультимножества и парные сравнения

Мультимножество – это множество с повторяющимися элементами A , порожденное универсальным множеством всех возможных вариантов $X = \{x_1, x_2, \dots\}$, где $A = \{k_A(x) \circ x \mid x \in X, k_A \in Z_+\}$, $k_A : X \rightarrow Z_+ = \{0, 1, 2, \dots\}$ – функция кратности, определяющая число экземпляров элемента x в мультимножестве A . Примеры мультимножеств: совокупность многопризнаковых объектов, коллекция графических символов, массив текстовых документов [1]. Аппарат теории мультимножеств успешно применяется для решения практических задач [4-7]. На основе мультимножеств разработаны методы обработки экспертных оценок:

- метод АРАМИС (Агрегирование и Ранжирование Альтернатив около Многопризнаковых Идеальных Ситуаций) позволяет обрабатывать как числовые, так и вербальные оценки. Метод основан на представлении сравниваемых вариантов в виде мультимножеств оценок, которые упорядочиваются по относительной близости к лучшему и худшему вариантам (опорным ситуациям) [1];

- метод АИР (Агрегирование Индивидуальных Ранжировок) предназначен для обработки упорядочений вариантов. Основан на построении мультимножеств характеризующих все варианты и последовательном исключении из рассмотрения вариантов, которые расположены на минимальном расстоянии от худшей ситуации [4].

Отметим, что применение мультимножеств позволяет формулировать обоснование выбора для лица, принимающего решение. В виду высокой эффективности методов (продемонстрированной при решении практических задач),

используемых данный математический аппарат, были решено применить его для обработки парных сравнений.

Предположим, что все суждения каждого эксперта согласовны и представлены в виде транзитивных матриц парных сравнений с использованием как числовых, так и вербальных значений. В случае несогласованности и/или нетранзитивности экспертных суждений предлагается применять к ним процедуру их выявления (устранения), описанную в работе [5].

Согласно методологии применения мультимножеств для начала необходимо определить шкалы оценивания по каждому критерию. Число градаций должно быть нечетным, например:

- « A_i лучше A_j », « A_i эквивалентна A_j », « A_i хуже A_j »;
- « A_i намного лучше A_j », « A_i лучше A_j », « A_i эквивалентна A_j », « A_i хуже A_j », « A_i намного хуже A_j ».

Затем на основании всех матриц парных сравнений построим мультимножества, характеризующие каждый вариант. Построим мультимножество характеризующее вариант A_i :

$A_i = \{k_{A_i}(\text{«}A_i \text{ лучше } A_j \text{ по 1-му критерию}\text{»}) \circ \text{«}A_i \text{ лучше } A_j \text{ по 1-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ эквивалентна } A_j \text{ по 1-му критерию}\text{»}) \circ \text{«}A_i \text{ эквивалентная } A_j \text{ по 1-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ хуже } A_j \text{ по 1-му критерию}\text{»}) \circ \text{«}A_i \text{ хуже } A_j \text{ по 1-му критерию}\text{»}; k_{A_i}(\text{«}A_i \text{ намного лучше } A_j \text{ по 2-му критерию}\text{»}) \circ \text{«}A_i \text{ намного лучше } A_j \text{ по 2-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ лучше } A_j \text{ по 2-му критерию}\text{»}) \circ \text{«}A_i \text{ лучше } A_j \text{ по 2-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ эквивалентна } A_j \text{ по 2-му критерию}\text{»}) \circ \text{«}A_i \text{ эквивалентна } A_j \text{ по 2-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ хуже } A_j \text{ по 2-му критерию}\text{»}) \circ \text{«}A_i \text{ хуже } A_j \text{ по 2-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ намного хуже } A_j \text{ по 2-му критерию}\text{»}) \circ \text{«}A_i \text{ намного хуже } A_j \text{ по 2-му критерию}\text{»}; \dots; k_{A_i}(\text{«}A_i \text{ лучше } A_j \text{ по } m\text{-му критерию}\text{»}) \circ \text{«}A_i \text{ лучше } A_j \text{ по } m\text{-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ эквивалентна } A_j \text{ по } m\text{-му критерию}\text{»}) \circ \text{«}A_i \text{ эквивалентная } A_j \text{ по } m\text{-му критерию}\text{»}, k_{A_i}(\text{«}A_i \text{ хуже } A_j \text{ по } m\text{-му критерию}\text{»}) \circ \text{«}A_i \text{ хуже } A_j \text{ по } m\text{-му критерию}\text{»}\}$, где $k_{A_i}(x_s^{es})$ – число экспертов, давших объекту A_i оценку $x_s^{es} \in X_s, s=1, \dots, m$.

Применение такой структуры мультимножеств позволяет обрабатывать как числовые, так и вербальные оценки в парных сравнениях вариантов.

Далее по аналогии с методом АРАМИС можно построить мультимножества, характеризующие опорные ситуации – лучший и худший варианты. Затем оценить показатель относительной близости к лучшему варианту в метрическом пространстве Петровского.

4. Пример

В качестве примера рассмотрим следующую задачу: двум экспертам предложено оценить 4 альтернативы по трем критериям с использованием вербальных оценок в парных сравнениях. Были получены матрицы парных сравнений:

1 эксперт

C_1	A_1	A_2	A_3	A_4
A_1	э	х	э	х
A_2	л	э	х	л
A_3	э	л	э	х
A_4	л	х	л	э

C_2	A_1	A_2	A_3	A_4
A_1	э	э	л	л
A_2	э	э	х	х
A_3	х	л	э	э
A_4	х	л	э	э

C_3	A_1	A_2	A_3	A_4
A_1	э	л	х	х
A_2	х	э	э	х
A_3	л	э	э	э
A_4	л	л	э	э

2 эксперт

C_1	A_1	A_2	A_3	A_4
A_1	э	л	х	л
A_2	х	э	х	х
A_3	л	л	э	х
A_4	х	л	л	э

C_2	A_1	A_2	A_3	A_4
A_1	э	э	э	л
A_2	э	э	э	х
A_3	э	э	э	э
A_4	х	л	э	э

C_3	A_1	A_2	A_3	A_4
A_1	э	л	э	х
A_2	х	э	л	э
A_3	э	х	э	х
A_4	л	э	л	э

Построим мультимножества, характеризующие все варианты:

$A_1 = \{0^* \ll A_1 \text{ лучше } A_1 \text{ по } C_1 \gg, 2^* \ll A_1 \text{ эквивалентна } A_1 \text{ по } C_1 \gg, 0^* \ll A_1 \text{ хуже } A_1 \text{ по } C_1 \gg;$
 $1^* \ll A_1 \text{ лучше } A_2 \text{ по } C_1 \gg, 0^* \ll A_1 \text{ эквивалентна } A_2 \text{ по } C_1 \gg, 1^* \ll A_1 \text{ хуже } A_2 \text{ по } C_1 \gg;$
 $0^* \ll A_1 \text{ лучше } A_3 \text{ по } C_1 \gg, 1^* \ll A_1 \text{ эквивалентна } A_3 \text{ по } C_1 \gg, 1^* \ll A_1 \text{ хуже } A_3 \text{ по } C_1 \gg;$
 $1^* \ll A_1 \text{ лучше } A_4 \text{ по } C_1 \gg, 0^* \ll A_1 \text{ эквивалентна } A_4 \text{ по } C_1 \gg, 1^* \ll A_1 \text{ хуже } A_4 \text{ по } C_1 \gg;$
 $0^* \ll A_1 \text{ лучше } A_1 \text{ по } C_2 \gg, 2^* \ll A_1 \text{ эквивалентна } A_1 \text{ по } C_2 \gg, 0^* \ll A_1 \text{ хуже } A_1 \text{ по } C_2 \gg;$
 $0^* \ll A_1 \text{ лучше } A_2 \text{ по } C_2 \gg, 2^* \ll A_1 \text{ эквивалентна } A_2 \text{ по } C_2 \gg, 1^* \ll A_1 \text{ хуже } A_2 \text{ по } C_2 \gg;$
 $1^* \ll A_1 \text{ лучше } A_3 \text{ по } C_2 \gg, 1^* \ll A_1 \text{ эквивалентна } A_3 \text{ по } C_2 \gg, 0^* \ll A_1 \text{ хуже } A_3 \text{ по } C_2 \gg;$
 $2^* \ll A_1 \text{ лучше } A_4 \text{ по } C_2 \gg, 0^* \ll A_1 \text{ эквивалентна } A_4 \text{ по } C_2 \gg, 0^* \ll A_1 \text{ хуже } A_4 \text{ по } C_2 \gg;$
 $0^* \ll A_1 \text{ лучше } A_1 \text{ по } C_3 \gg, 2^* \ll A_1 \text{ эквивалентна } A_1 \text{ по } C_3 \gg, 0^* \ll A_1 \text{ хуже } A_1 \text{ по } C_3 \gg;$
 $2^* \ll A_1 \text{ лучше } A_2 \text{ по } C_3 \gg, 0^* \ll A_1 \text{ эквивалентна } A_2 \text{ по } C_3 \gg, 0^* \ll A_1 \text{ хуже } A_2 \text{ по } C_3 \gg;$
 $0^* \ll A_1 \text{ лучше } A_3 \text{ по } C_3 \gg, 1^* \ll A_1 \text{ эквивалентна } A_3 \text{ по } C_3 \gg, 1^* \ll A_1 \text{ хуже } A_3 \text{ по } C_3 \gg;$
 $0^* \ll A_1 \text{ лучше } A_4 \text{ по } C_3 \gg, 0^* \ll A_1 \text{ эквивалентна } A_4 \text{ по } C_3 \gg, 2^* \ll A_1 \text{ хуже } A_4 \text{ по } C_3 \gg;\}$

Для последующих альтернатив мультимножества представлены в табличном виде:

A_1		лучше	эквивалентна	хуже
C_1	A_1	0	2	0
	A_2	1	0	1
	A_3	0	1	1
	A_4	1	0	1
C_2	A_1	0	2	0
	A_2	0	2	0
	A_3	1	1	0
C_3	A_1	0	2	0
	A_2	2	0	0
	A_3	0	1	1
	A_4	0	0	2

A_2		лучше	эквивалентна	хуже
C_1	A_1	1	0	1
	A_2	0	2	0
	A_3	0	0	2
	A_4	1	0	1
C_2	A_1	0	2	0
	A_2	0	2	0
	A_3	0	1	1

	A_4	0	0	2
C_3	A_1	0	0	2
	A_2	0	2	0
	A_3	1	1	0
	A_4	0	1	1

A_3		лучше	эквивалентна	хуже
C_1	A_1	1	1	0
	A_2	2	0	0
	A_3	0	2	0
	A_4	0	0	2
C_2	A_1	0	1	1
	A_2	1	1	0
	A_3	0	2	0
	A_4	0	2	0
C_3	A_1	1	1	0
	A_2	0	1	1
	A_3	0	2	0
	A_4	0	1	1

A_4		лучше	эквивалентна	хуже
C_1	A_1	1	0	1
	A_2	1	0	1
	A_3	2	0	0
	A_4	0	2	0
C_2	A_1	0	0	2
	A_2	2	0	0

	A_3	0	2	0
	A_3	0	2	0
C_3	A_1	2	0	0
	A_2	1	1	0
	A_3	1	1	0
	A_3	0	2	0

Построим мультимножества, характеризующие опорные ситуации:

A_1		лучше	эквивалентна	хуже
C_1	A_1	0	2	0
	A_2	1	0	1
	A_3	0	1	1
	A_4	1	0	1
C_2	A_1	0	2	0
	A_2	0	2	0
	A_3	1	1	0
	A_4	2	0	0
C_3	A_1	0	2	0
	A_2	2	0	0
	A_3	0	1	1
	A_4	0	0	2

A_4		лучше	эквивалентна	хуже
C_1	A_1	1	0	1
	A_2	1	0	1
	A_3	2	0	0
	A_4	0	2	0
C_2	A_1	0	0	2
	A_2	2	0	0
	A_3	0	2	0
	A_4	0	2	0
C_3	A_1	2	0	0
	A_2	1	1	0
	A_3	1	1	0
	A_4	0	2	0

A_2		лучше	эквивалентна	хуже
C_1	A_1	1	0	1
	A_2	0	2	0
	A_3	0	0	2
	A_4	1	0	1
C_2	A_1	0	2	0
	A_2	0	2	0
	A_3	0	1	1
	A_4	0	0	2
C_3	A_1	0	0	2
	A_2	0	2	0
	A_3	1	1	0
	A_4	0	1	1

A_3		лучше	эквивалентна	хуже
C_1	A_1	1	1	0
	A_2	2	0	0
	A_3	0	2	0
	A_4	0	0	2
C_2	A_1	0	1	1
	A_2	1	1	0
	A_3	0	2	0
	A_4	0	2	0
C_3	A_1	1	1	0
	A_2	0	1	1
	A_3	0	2	0
	A_4	0	1	1

Построим мультимножества, характеризующие опорные ситуации:

A_+		лучше	эквивалентна	хуже
C_1	A_1	2	0	0
	A_2	2	0	0
	A_3	2	0	0
	A_4	2	0	0
C_2	A_1	2	0	0
	A_2	2	0	0
	A_3	2	0	0
	A_4	2	0	0
C_3	A_1	2	0	0
	A_2	2	0	0
	A_3	2	0	0
	A_4	2	0	0

A_+		лучше	эквивалентна	хуже
C_1	A_1	0	0	2
	A_2	0	0	2
	A_3	0	0	2
	A_4	0	0	2
C_2	A_1	0	0	2
	A_2	0	0	2
	A_3	0	0	2
	A_4	0	0	2
C_3	A_1	0	0	2
	A_2	0	0	2
	A_3	0	0	2
	A_4	0	0	2

Определим расстояния между альтернативами и опорными ситуациями:

$d(A_+, A_1)$		лучше	эквивалентна	хуже	
C_1	A_1	2-0	2	0	4
	A_2	2-1	0	1	2
	A_3	2-0	1	1	4
	A_4	2-1	0	1	2
C_2	A_1	2-0	2	0	4
	A_2	2-0	2	0	4
	A_3	2-1	1	0	2
	A_4	2-2	0	0	0
C_3	A_1	2-0	2	0	4
	A_2	2-2	0	0	2
	A_3	2-0	1	1	4
	A_4	2-0	0	2	4
Итого					36

$d(A_+, A_2)$		лучше	эквивалентна	хуже	
C_1	A_1	2-1	0	1	2
	A_2	2-0	2	0	4
	A_3	2-0	0	2	4
	A_4	2-1	0	1	2
C_2	A_1	2-0	2	0	4
	A_2	2-0	2	0	4
	A_3	2-0	1	1	4
	A_4	2-0	0	2	4
C_3	A_1	2-0	0	2	4
	A_2	2-0	2	0	4
	A_3	2-1	1	0	2
	A_4	2-0	1	1	4
Итого					42

$d(A_+, A_3)$		лучше	эквивалентна	хуже	
C_1	A_1	2-1	1	0	2
	A_2	2-2	0	0	0
	A_3	2-0	2	0	4
	A_4	2-0	0	2	4
C_2	A_1	2-0	1	1	4
	A_2	2-1	1	0	2
	A_3	2-0	2	0	4
	A_4	2-0	2	0	4
C_3	A_1	2-1	1	0	2
	A_2	2-0	1	1	4
	A_3	2-0	2	0	4
	A_4	2-0	1	1	4
Итого					38

$d(A_+, A_4)$		лучше	эквивалентна	хуже	
C_1	A_1	2-1	0	1	2
	A_2	2-1	0	1	2
	A_3	2-2	0	0	0
	A_4	2-0	2	0	4
C_2	A_1	2-0	0	2	4
	A_2	2-2	0	0	0
	A_3	2-0	2	0	2
	A_4	2-0	2	0	4
C_3	A_1	2-2	0	0	0
	A_2	2-1	1	0	2
	A_3	2-1	1	0	2
	A_4	2-0	2	0	4
Итого					26

$d(A_-, A_1)$		лучше	эквивалентна	хуже	
C_1	A_1	0	2	2-0	4
	A_2	1	0	2-1	2
	A_3	0	1	2-1	2
	A_4	1	0	2-1	2
C_2	A_1	0	2	2-0	4
	A_2	0	2	2-0	4
	A_3	1	1	2-0	4
	A_4	2	0	2-0	4
C_3	A_1	0	2	2-0	4
	A_2	2	0	2-0	4
	A_3	0	1	2-1	2
	A_4	0	0	2-2	0
Итого					36

$d(A_1, A_2)$		лучше	эквивалентна	хуже	
C ₁	A ₁	1	0	2-1	2
	A ₂	0	2	2-0	4
	A ₃	0	0	2-2	0
	A ₄	1	0	2-1	2
C ₂	A ₁	0	2	2-0	4
	A ₂	0	2	2-0	4
	A ₃	0	1	2-1	2
	A ₄	0	0	2-2	0
C ₃	A ₁	0	0	2-2	0
	A ₂	0	2	2-0	4
	A ₃	1	1	2-0	4
	A ₄	0	1	2-1	2
Итого					28

$d(A_1, A_3)$		лучше	эквивалентна	хуже	
C ₁	A ₁	1	1	2-0	2
	A ₂	2	0	2-0	4
	A ₃	0	2	2-0	4
	A ₄	0	0	2-2	0
C ₂	A ₁	0	1	2-1	2
	A ₂	1	1	2-0	4
	A ₃	0	2	2-0	4
	A ₄	0	2	2-0	4
C ₃	A ₁	1	1	2-0	4
	A ₂	0	1	2-1	2
	A ₃	0	2	2-0	4
	A ₄	0	1	2-1	2
Итого					36

$d(A_1, A_4)$		лучше	эквивалентна	хуже	
C ₁	A ₁	1	0	2-1	2
	A ₂	1	0	2-1	2
	A ₃	2	0	2-0	4
	A ₄	0	2	2-0	4
C ₂	A ₁	0	0	2-2	0
	A ₂	2	0	2-0	4
	A ₃	0	2	2-0	4
	A ₄	0	2	2-0	4
C ₃	A ₁	2	0	2-0	4
	A ₂	1	1	2-0	4
	A ₃	1	1	2-0	4
	A ₄	0	2	2-0	4
Итого					40

Рассчитаем величин показателя относительной близости вариантов к лучшей ситуации:

	A_+	A_-	l
A_1	36	36	0,5
A_2	42	28	0,6
A_3	38	36	0,51
A_4	26	40	0,39

Минимальное значение показателя относительной близости к лучшей ситуации у четвертого варианта, который будет предъявлен лицу, принимающему решение.

5. Заключение

В работе рассмотрена возможность применения аппарата мультимножеств в метрическом пространстве Перовского к обработке экспертных суждений, представленных в виде парных сравнений (матриц парных сравнений). Использование данного математического аппарата позволяет решать задачу группового многокритериального выбора и предоставлять лицу, принимающему решение, обоснование для выбора лучшего варианта.

Работа выполнена при частичной финансовой поддержке РФФИ (проект №16-07-0112).

Литература

19. Петровский А. Б. Теория принятия решений. — М.: Издательский центр «Академия», 2009.
20. Малтугуева Г.С., Петровский А.Б. Многометодное групповое многокритериальное оценивание экологически мероприятий // Шестая международная конференция «Системный анализ и информационные технологии» САИТ – 2015 (15-20 июня 2015 г., г. Светлогорск, Россия): Труды конференции. В 2-х т. – Т.2. М.: ИСА РАН. – С. 72-80.
21. Миркин Б.Г. Проблема группового выбора.- М.: Наука, 1974.
22. Малтугуева Г.С., Юрин А.Ю. Агрегирование предпочтений в группах. Метод и программное средство. – Berlin: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2014.
23. Ашихмин И.В., Ройзензон Г.В. Выбор лучшего объекта на основе парных сравнений на подмножестве критериев // Методы принятия решений. М.: Едиториал УРСС, 2001. – С. 51-71.

РАЗРАБОТКА АЛГОРИТМА ОБРАБОТКИ ГРУППОВЫХ МНОГОКРИТЕРИАЛЬНЫХ ЭКСПЕРТНЫХ ОЦЕНОК НА ОСНОВЕ МЕТРИЧЕСКИХ ПРОСТРАНСТВ МУЛЬТИМНОЖЕСТВ⁴

Проничкин С.В.

ФИЦ ИУ РАН

Москва, пр-т 60-летия Октября, 9, pronichkin@mail.ru

Аннотация

В работе рассматривается новый подход к обработке вербальных оценок, в которых выражаются мнения экспертов по многим критериям. Предлагается алгоритм вербального анализа экспертных оценок многопризнаковых объектов для выделения подгрупп экспертов с согласованными мнениями. В качестве методологической основы алгоритма используются мультимножества, которые наиболее подходят для структуризации и анализа совокупности объектов, которые описываются многими вербальными признаками, а также могут присутствовать в нескольких версиях. Для вычисления коэффициента согласованности предлагается использовать показатели кратности мультимножеств. Также предлагается использовать совместно значения коэффициентов согласованности для оценки согласованности подгрупп экспертов по одному признаку, и стремиться к суммарной максимизации коэффициентов согласованности по всем признакам. Дана оценка сложности предлагаемого алгоритма. Представлены результаты вычислений для задачи выбора наилучшей технологии обработки декоративного камня.

Ключевые слова: мультимножества; принятие решений; согласованность мнений; уровень готовности технологии.

1. Введение

Имеется достаточно широкий круг задач исследования сложных систем и процессов, где изучаемые объекты характеризуются многими признаками, по которым получены вербальные оценки от множества экспертов. Основной целью обработки коллективных экспертных оценок является получение обобщенных данных и выявление новой информации, содержащейся в скрытой форме в экспертных оценках. Коллектив экспертов обычно расходится во мнении. При отсутствии согласованности экспертов целесообразно разбивать их на группы сходных по мнению. В связи с этим возникает важная задача выделения подгрупп экспертов с согласованными мнениями.

⁴ Статья подготовлена при поддержке РФФИ, гранты №16-07-00138 и №16-36-00080.

2. Математическая постановка задачи

Математическая постановка задачи выглядит следующим образом.

Пусть m экспертов произвели оценку n объектов по s критериям. Результаты оценки представим в виде величин x_{ijl} , где i - номер эксперта ($i=1, \dots, m$), j - номер объекта ($j=1, \dots, n$), l - номер критерия (признака) ($l=1, \dots, s$). Известна относительная важность критериев w_l , $l=1, \dots, s$, $\sum_{l=1}^s w_l = 1$.

Для получения разбиения исходной группы экспертов на подгруппы со сходными мнениями для заданного критерия принято использовать коэффициенты согласованности мнений [1, 2, 3], которые вычисляются на рангах, и позволяют оценивать согласованность мнений экспертов в группе.

В работе [4] предложен коэффициент множественной качественной согласованности (1), который не основан на рангах.

$$W_k = 1 - \frac{\sum_{j=1}^J \sum_{l_1=1}^L \sum_{l_2=1}^L |x_{l_1 j} - x_{l_2 j}|}{J \cdot L \cdot (L-1) \cdot (K-1)}, \quad (1)$$

где $x_{l_1 j}$, $x_{l_2 j}$ - значение j -го критерия объекта по результатам оценки экспертом l_1 , l_2 ;

L - число экспертов;

J - число критериев;

K - основание качества системы признаков.

Областью значения коэффициента (1) является отрезок $[0, 1]$. Значению 0 соответствует максимальная несогласованность, значению 1 полная согласованность.

Коэффициент (1) не свободен от недостатков, поскольку не учитывает распределение оценок экспертов от одного объекта к другому и основан исключительно на числовых экспертных оценках по критериям, полученных в шкале интервалов.

Числовое измерение качественных показателей, несмотря на кажущуюся простоту и очевидность, мало пригодно для работы с качественными характеристиками, которыми описываются объекты, так как содержит целый ряд методологических дефектов, хорошо известных специалистам.

Практически невозможно априори назначить количественные шкалы оценок, сопоставив качественным факторам какие-либо числа так, чтобы они «правильно» выражали плохо формализуемые свойства объектов и одинаково понимались разными людьми. Тем более, числовые оценки неприменимы для измерения непорядковых показателей. Более того, было строго показано, что использование даже по одним и тем же критериям порядковых шкал, имеющих разные числовые шкалы оценок может привести к совершенно разным упорядочениям и разбиениям на классы исходной совокупности объектов. При этом не существует никакой содержательной аргументации в пользу выбора той или иной шкалы. Тем самым, только путем произвольного изменения градаций оценок на шкалах можно получить любую ранжировку и любую классификацию многопризнаковых объектов. Наконец, при использовании шкал числовых оценок появляется искушение сформулировать один «простой» итоговый показатель, агрегирующий частные оценки (обычно в виде суммы,

взвешенной суммы или некоторой усредненной оценки), по величине которого и проводится сравнение объектов. В этом случае происходит смешивание разнородных показателей, важных и не важных факторов, оценок разных экспертов, что не позволяет выделить наиболее значимые для выбора показатели.

Также в работе [4] не предложен механизм оценки значимости предлагаемого коэффициента множественной качественной согласованности, т.е. строго обоснования статистической достоверности полученного разбиения. Важно отметить, что не предложен механизм выделения подгрупп экспертов с согласованными мнениями, в случае если коэффициент множественной качественной согласованности близок к нулю. В тоже время, в работе [5] предлагается алгоритм, в котором последовательно исключаются подгруппы с согласованными мнениями из исходной группы, и тем самым исходная группа разбивается на подгруппы с согласованными мнениями для заданного критерия.

Такой подход не свободен от недостатков:

1. Поиск подгруппы экспертов с согласованными мнениями прекращается, как только максимальное значение коэффициента согласованности для текущего числа экспертов в подгруппе окажется значимым. Это приводит к тому, что подгруппа экспертов с согласованными мнениями, полученная первой, будет больше чем остальные подгруппы, и в общем случае разбиение исходной группы экспертов на подгруппы с согласованными мнениями не будет оптимальным в смысле совместной максимизации коэффициентов согласованности для подгрупп экспертов с согласованными мнениями для заданного критерия.

2. Учитывается только значение коэффициента согласованности для подгрупп экспертов, не учитывается диапазон изменения значений коэффициента, рассеяния его значений для заданного критерия, что приводит к неудовлетворительным решениям, т.к. согласованность одной подгруппы экспертов может сильно отличаться от согласованности другой.

3. Разбиение исходной группы экспертов на подгруппы с согласованными мнениями осуществляется для заданного (одного) критерия. В общем случае получается, что для каждого объекта в зависимости от критерия набор экспертов её оценивших будет разным, причем количество экспертов в подгруппе со сходными мнениями для одного критерия может сильно отличаться от количества экспертов в подгруппе со сходными мнениями для другого критерия.

3. Алгоритм обработки экспертных оценок

В качестве методологической основы алгоритма обработки экспертных оценок предлагается использовать мультимножества [6, 7, 8]. Теоретическая модель мультимножества наиболее подходит для структуризации и анализа совокупности объектов, которые описываются многими вербальными признаками, а также могут присутствовать в нескольких версиях. Оценка объекта представляется как совокупность групп элементов $A_i = \{r_{Ai}(x) \circ x | x \in X, r_{Ai} \in Z_+\}$, где $r_{Ai} : X \rightarrow Z_+ = \{0,1,2,\dots\}$ - функция кратности мультимножества, определяющая число вхождений элемента $x_j \in X$ в мультимножество A_i , по результатам i -ой экспертизы. Таким образом, для вычисления коэффициента согласованности предлагается использовать показатели кратности мультимножеств. Также предлагается использовать совместно значения коэффициентов согласованности для подгрупп экспертов, и стремиться к их суммарной максимизации в то же время желательно, чтобы согласованность в подгруппах не сильно отличалась. Каждый объект оценивается каждым экспертом по многим

критериям, поэтому целесообразно выделение подгрупп экспертов с согласованными мнениями по всем критериям с учетом их важности.

Для преодоления первого и второго недостатков существующих алгоритмов предлагается построить целевую функцию $f(W_i^{(t)})$. Которая представляет собой объединение (свертку) среднего значения коэффициентов согласованности и среднего квадратического отклонения коэффициентов согласованности для заданного критерия l , где t - число подгрупп экспертов.

Необходимо найти такое $W_i^{(t)}$, при котором среднее значение коэффициентов согласованности будет больше, а значение среднего квадратического отклонения коэффициентов согласованности будет меньше. Варьирование значений параметра $\lambda_l \in [0, 1]$ изменяет свойства целевой функции $f(W_i^{(t)})$ для l -го критерия, позволяя в большей или в меньшей степени в зависимости от величины λ_l учитывать величины среднего значения коэффициентов согласованности и среднего квадратического отклонения коэффициентов согласованности, включенных в целевую функцию.

Для преодоления третьего недостатка предлагается построить целевую функцию. Проведем объединение (свертку) среднего значения и среднего квадратического отклонения значений целевой функции $f(W_i^{(t)})$ с учетом важности критериев. В результате получаем следующую целевую функцию:

$$F(W^{(t)}) = (1 - \lambda) \frac{\sum_{i=1}^I w_i f(W_i^{(t)})}{I} - \lambda \sqrt{\frac{\sum_{i=1}^I w_i^2 (\bar{f}(W_i^{(t)}) - f(W_i^{(t)}))^2}{I}}, \quad (2)$$

где $W^{(t)} = (W_1^{(t)}, \dots, W_I^{(t)})$ – векторы коэффициентов согласованности, полученные для разбиения исходной группы экспертов на t подгрупп $\mathcal{E}^{(t)} = (\mathcal{E}_1, \dots, \mathcal{E}_{c_1}, \dots, \mathcal{E}_{c_{t-1}+1}, \dots, \mathcal{E}_{c_t})$;

$\bar{f}(W_i^{(t)})$ – средневзвешенное значение функций $f(W_i^{(t)})$;

w_i – важность критериев, $i = 1, \dots, I$, $\sum_{i=1}^I w_i = 1$;

λ – параметр (весовой коэффициент).

Задача выделения подгрупп экспертов с согласованными мнениями формализована в виде целевой функции (3):

$$F(W_{opt}^{(t)}) = \max_{t \in \{t: W^{(t)} \in W_{знач}^{(t)}\}} F(W^{(t)}), \quad (3)$$

где $W_{знач}^{(t)}$ – множество значимых коэффициентов согласованности, $W_{знач}^{(t)} = \{W_i(c_u) : W_i(c_u) \geq W_{Ti}(c_u), u = 1, \dots, t, i = 1, \dots, I\}$;

$W_{Ti}(c_u)$ – теоретическое значение коэффициента согласованности для подгруппы экспертов u .

Необходимо найти решение $\mathcal{E}_{opt}^{(t)}$, которому соответствует $W_{opt}^{(t)}$, из условия (3).

Формальное описание алгоритма решения поставленной задачи (3):

1. Задать $p_{ош}(i)$ – вероятность ошибки, $\lambda_i, i = 1, \dots, I$.
2. $\lambda = 0, t_1 = 1$.
3. Полные матрицы кратностей мультимножеств $\|r_{dp}\|_i, d = 1, \dots, m, p = 1, \dots, P, i = 1, \dots, I$ последовательно в разных сочетаниях делить на t_1 групп строк, строк в

группе $c_u = 2, \dots, m - 2(t_1 - 1)$, $u = 1, \dots, t_1$, если $m - 2t_1 \geq 1$, то $\sum_{u=1}^{t_1} c_u = m - 1, m$, иначе

$\sum_{u=1}^{t_1} c_u = m$. Для каждого набора строк вычислить $W_i^{(t_1)} = (W_i(c_1), \dots, W_i(c_{t_1}))$, $i = 1, \dots, I$ и сравнить $W_i(c_u)$ с $W_{T_i}(c_u)$, если $W_i(c_u) \geq W_{T_i}(c_u)$, $u = 1, \dots, t_1$, $i = 1, \dots, I$, то вычислить $F(W^{(t_1)})$. Найти $F(W^{(t_1)})$ с максимальным значением $F(W_{onm}^{(t_1)})$.

4. $t_1 = t_1 + 1$.

5. Проверить условие $t_1 \leq \left\lfloor \frac{m}{2} \right\rfloor$, условие выполнено перейти к п. 3.

6. Найти $F(W_{onm}^{(t_1)})$ с максимальным значением $F(W_{onm}^{(t)})$. Запомнить λ , t , $W_{onm}^{(t)} = (W_1^{(t)*}, \dots, W_I^{(t)*})$ и соответствующие матрицей кратностей мультимножеств $\|r_{dp}^1\|_i$, $d = 1, \dots, c_1, \dots, \|r_{dp}^t\|_i$, $d = 1, \dots, c_t$, $p = 1, \dots, P$, $i = 1, \dots, I$ с согласованными мнениями экспертов.

7. $\lambda = \lambda + 0.1$.

8. Проверить условие $\lambda \leq 1$, условие выполнено перейти к п. 3.

9. В случае если решение не найдено перейти к п. 1.

В предложенном алгоритме значения параметров λ_l , $l = 1, \dots, s$ можно задавать непосредственно перед вычислением $F(W^{(t)})$ см. п. 3. В качестве примера шаг изменения параметра λ выбран 0.1. Величину шага изменения параметра λ , h_λ задает ЛПР, затем ЛПР изучает полученные решения (3) при $\lambda = \{0; h_\lambda; \dots; 1\}$ и выбирает λ соответствующие его представлению о качестве решения.

Оценка сложности представленного выше алгоритма является экспоненциальной $O\left(\sum_{t_1=1}^{\left\lfloor \frac{m}{2} \right\rfloor} C_m^{c_1} \cdot C_{m-c_1}^{c_2} \cdot \dots \cdot C_{m-c_1-\dots-c_{t_1-1}}^{c_{t_1}}\right)$, где $C_m^{c_1}$ – число сочетаний из m

различных элементов по c_1 элементов, порядок элементов не имеет значение. Несмотря на то, что оценка сложности экспоненциальная, тем не менее, имеется эмпирическое правило для количества экспертов в группе, оно должно изменяться от 2 до 12. В указанных пределах алгоритм дает точное решение без ощутимых временных затрат. Конкретные времена расчёта в указанных пределах составляют от 0,136 до 4,271 сек.

4. Результаты вычислений для задачи выбора наилучшей технологии

Современная камнеобрабатывающая промышленность применяет для обработки горных пород различные технологии, которые можно разделить на два больших класса [9, 10, 11]: механические и физико-механические.

В настоящее время наибольшее распространение получили технологии на основе механического способа обработки камня, такие как скалывание, резание и ударное разрушение, которые до настоящего времени являются наиболее распространенными.

Среди физико-технических способов обработки камня нашли широкое применение технологии обработки терморезактивными газовыми горелками, токами высокой частоты, высокоскоростной водяной струей и плазменного разрушения пород.

В настоящее время активно разрабатываются новые технологии [12, 13, 14] обработки горных пород. Задача выбора технологии является многокритериальной и означает, что в ходе её реализации соблюдаются все необходимые ограничения технического, экологического, социального, финансового и другого характера. Показатели оценки эффективности технологии могут представлять собой набор индикаторов различного характера. Они могут выражаться как в качественной, так и количественной форме. Однако, только использование качественных показателей позволяет получить всестороннюю экспертную оценку.

Качественный подход решения задачи выбора технологии позволяет преодолеть сложность, которая заключается в отсутствии методов получения объективных количественных измерителей, таких показателей как: вклад разрабатываемой технологии в основное направление предприятия; вероятность успеха в реализации технологии; приращение уровня готовности технологии за планируемый период. Такая задача может быть решена введением целостной системы оценки уровней готовности технологий, включающей не только собственно оценку, но и вопросы системной интеграции технологии.

Таким образом, преимуществом качественных показателей является возможность получения более широкой оценки за счет того, что оценка формулируется не в виде одного или нескольких количественных характеристик, а в виде качественного описания. В частности, в мировой практике нашла широкое применение шкала уровня готовности технологий [15, 16], которая представляет собой набор критериев для оценки технологии с точки зрения её внедрения, начиная от идеи и заканчивая готовым прототипом. Шкала уровня готовности позволяет последовательно сравнить степень готовности различных технологий по таким критериям как: степень поддержки программы разработки технологии; результативность теоретических и эмпирических проектных решений; полнота базовых элементов технологии; возможности совершенствования технологии и др. Причем на любом из уровней готовности технологии используются коллективные экспертные оценки.

Предлагаемый алгоритм применялся для решения задачи выбора наилучшей технологии для декоративной обработки камня. Оценка производилась с привлечением семи экспертов. Результаты вычислений зависят от выбора параметров см. рисунок 1.

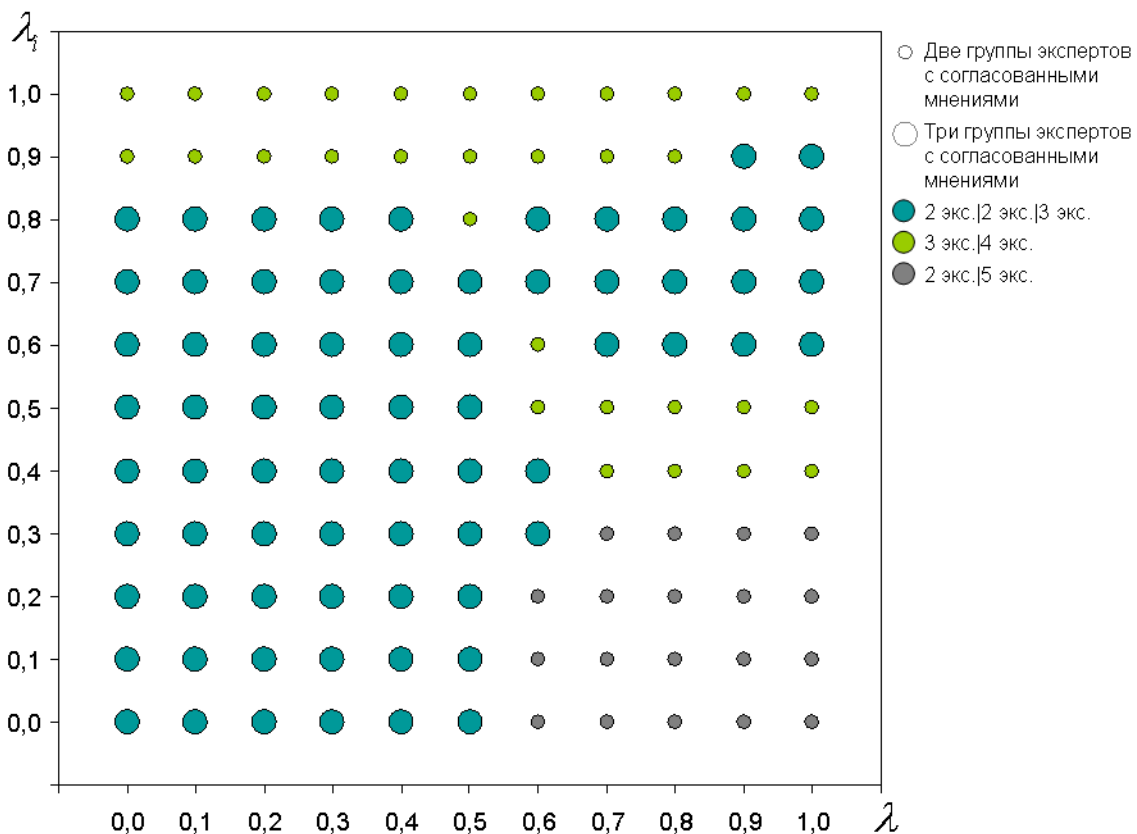


Рис. 1. – Зависимость разбиения (группы экспертов с согласованными мнениями) от значений параметров

Уменьшение значения λ (λ_i) приводит к увеличению количества групп и соответственно к их небольшому составу. Увеличение значения λ (λ_i) дает обратный эффект при этом приходится менять уровень значимости (увеличивать) т.к. коэффициенты согласованности становятся незначимыми для выделенных подгрупп.

Выбирая λ ЛПР получает результат соответствующий его представлению о качестве решения. Такая наглядность позволяет ЛПР изучать полученное множество решений, более ясно формулировать свои требования, сравнивая различные решения.

5. Заключение

Основной целью обработки коллективных экспертных оценок является получение обобщенных данных и выявление новой информации, содержащейся в скрытой форме в экспертных оценках. При отсутствии согласованности экспертов целесообразно разбивать их на группы сходных по мнению. Изучение мнений подгрупп экспертов с согласованными мнениями по многим критериям может помочь лицу, принимающему решение, сформулировать стратегии выбора, более адекватные реальности, сделать его решения более обоснованными и разумными.

В работе предложен новый подход к обработке вербальных оценок, в которых выражаются мнения экспертов по многим критериям. Разработан алгоритм вербального анализа экспертных оценок многопризнаковых объектов для выделения подгрупп экспертов с согласованными мнениями. В качестве методологической основы алгоритма используются мультимножества, которые наиболее подходят для структуризации и анализа совокупности объектов, которые описываются многими вербальными признаками, а также могут присутствовать в нескольких версиях. Для

вычисления коэффициента согласованности используются показатели кратности мультимножеств. Также предлагается использовать совместно значения коэффициентов согласованности для оценки согласованности подгрупп экспертов по одному признаку, и стремиться к суммарной максимизации коэффициентов согласованности по всем признакам.

Предлагаемый алгоритм применялся для решения задачи выбора наилучшей технологии для декоративной обработки камня. Применение предлагаемого алгоритма обработки экспертных оценок на основе метрических пространств мультимножеств позволяет более адекватно, чем в случае применения существующих подходов, структурировать и провести анализ совокупности технологий, которые описываются многими вербальными признаками и присутствуют в нескольких версиях.

Литература

1. Chena C., Barnhart H. Assessing agreement with intraclass correlation coefficient and concordance correlation coefficient for data with repeated measures. *Computational Statistics and Data Analysis*. Vol. 60, - 2013, P. 132–145.
2. Tsai M. Comparison of concordance correlation coefficient via variance components, generalized estimating equations and weighted approaches with model selection. *Computational Statistics and Data Analysis*. Vol. 82, - 2015, P. 47–58.
3. Adam F., Humphreys P. *Encyclopedia of Decision Making and Decision Support Technologies*. – NY.: Information Science Reference, 2008. – 1064 p.
4. Этезов Б.Б. Разработка моделей и алгоритмов оценки качества системы высшего образования: Дис... канд. техн. наук: 05.13.17 – М.: МГУП, 2006. – 234 с.
5. Рыков А.С. Модели и методы системного анализа. – М.: Издательство «РМ». 2005. – 352 с.
6. Петровский А.Б. Пространства множеств и мультимножеств. – М: Едиториал УРСС. 2003. – 246 с.
7. Петровский А.Б. Методы групповой классификации многопризнаковых объектов (часть 1). *Искусственный интеллект и принятие решений*, № 3, - 2009, С. 3-14.
8. Петровский А.Б. Методы групповой классификации многопризнаковых объектов (часть 2). *Искусственный интеллект и принятие решений*, № 4, - 2009, С. 3-14.
9. Singh B., Goel R. *Engineering Rock Mass Classification*. – NY: Wiley, 2011, P. 365.
10. Ulusay R. *The ISRM suggested methods for rock characterization, testing and monitoring*. - NY: Springer, 2015, P. 280.
11. Zhu H., Li X., Zhuang X. Recent advances of digitization in rock mechanics and rock engineering. *Journal of Rock Mechanics and Geotechnical Engineering*, Vol. 3, - 2011, P. 220-233.
12. Mikaeil R., Ozcelik Y., Yousefi R., Ataei M., Hosseini S. Ranking the sawability of ornamental stone using Fuzzy Delphi and multi-criteria decision-making techniques. *Int Journal of Rock Mechanics & Mining Sciences*. Vol. 58, - 2013, P. 118–126.
13. Mikaeil R., Ataei M., Hoseinie S. Predicting the production rate of diamond wire saws in carbonate rocks cutting. *Ind Diamond Rev*. Vol. 3, - 2014, P. 28–34.
14. Souza A., Pinheiro B., Holanda J. Processing of floor tiles bearing ornamental rock-cutting waste. *Journal of Materials Processing Technology*, Vol. 210, - 2015, P. 1898-1904.
15. Altunok T., Cakmak T. A technology readiness levels (TRLs) calculator software for systems engineering and technology management tool. *Advances in Engineering Software*, Vol. 41, - 2010, P. 769–778.

16. Straub J. In search of technology readiness level (TRL) 10. Aerospace Science and Technology, Vol. 46, - 2015, P. 312-320.

ГЕТЕРОГЕННАЯ ДИАГНОСТИКА АРТЕРИАЛЬНОЙ ГИПЕРТЕНЗИИ

Кириков И.А.

Калининградский филиал ФИЦ ИУ РАН
Россия, 236000, г. Калининград, Гостиная ул., д. 5, kfiriran@yandex.ru

Румовская С.Б.

Калининградский филиал ФИЦ ИУ РАН
Россия, 236000, г. Калининград, Гостиная ул., д. 5, sophiyabr@gmail.com

Аннотация

Целостное восприятие объекта диагностики остаётся феноменом высочайшего искусства и на практике превалирует гомогенный подход, рассматривающий объект диагностики, пациента, с позиции отдельных аспектов, составляющих и подсистем, в частности, в лечебно-профилактических учреждениях широкого профиля. Применение технологии «виртуальных консилиумов» повысит эффективность и качество индивидуальных диагностических решений, поскольку консультации будут получены как результат моделирования коллективного обсуждения, обладающего синергией, рассматривая объекты диагностики, пациентов, как гетерогенные системы.

1. Введение

Развитие медицинской информатики связано с работами Г.М. Адельсона-Вельского, С.А. Айвазяна, Н.М. Амосова, П.К. Анохина, Р.М. Баевского, Н. Бейли, Н.Н. Блохина, М.М. Бонгарда, А.С. Брудно, В.И. Бураковского, М.Н. Вайнцвайга, А.А. Вишневого, С.А. Гаспаряна, И.М. Гельфанда, А.А. Генкина, А.Н. Горбань, Ш.А. Губермана, Е.В. Гублера, В.С. Гурфинкеля, М.В. Донского, Е.С. Енюкова, Ю.И. Журавлева, В.П. Карп, А.И. Китова, Б.А. Кобринского, А.Н. Колмогорова, А.С. Кронрод, П.Е. Кунина, А.И. Курочкиной, Л. Ластеда, Г.С. Лбова, В.П. Леонова, Л.Д. Линдербрата, В.И. Лищука, С.Я. Мarmorштейна, Л.Д. Мешалкина, О.П. Минцера, Н.С. Мисюка, В.С. Переверзева-Орлова, Ю.М. Романовского, К.С. Симоняна, М.Б. Славина, А.Л. Сыркина, Г.А. Хая, М.Л. Цетлина, Д.С. Чернавского, D. Cullen, P. Fisher, L. Landucci, E.T. Lee, J.H. Van Vemmel и другими. Они были инициаторами рождения медицинской кибернетики. Строго говоря, само системное понимание проблем здравоохранения и даже сам термин «здравоохранение» появились именно в нашей стране и были восприняты всем миром. С именами вышеупомянутых ученых связаны первые работы:

1) по анализу больших объемов информации для повышения качества принятия решения относительно состояния больного [1, 2];

2) по прогнозированию изменения состояния больного в процессе развития заболевания и их исходов [3];

3) по контролю состояния больных в реальном масштабе времени, мониторинговых систем, в том числе с расчетом различных параметров, характеризующих патологический процесс [4]: здесь отдельными блоками стояли системы с дистанционным наблюдением за пациентом и с иерархической структурой сети ЭВМ – с локальными ЭВМ

для каждого пациента и центральной ЭВМ, обрабатывающей информацию и выдающей заключение [5].

Компьютеризированные системы в медицине ориентированы [6] на мониторинг состояния здоровья населения, консультативную поддержку в клинической медицине (диагностика, прогнозирование, лечение), переход к электронным историям болезни и амбулаторным медицинским картам, автоматизацию функциональной и лабораторной диагностики. Автоматизированные информационные системы больниц и поликлиник представляют собой иерархические системы четырех уровней: первый – медико-технологические системы, т.е. автоматизация диагностического обследования пациентов; второй – автоматизация рабочих мест врачей; третий – автоматизация лечебно-диагностических и управленческих систем подразделений (лабораторий и отделений); четвертый – автоматизированная информационная система лечебно-профилактического учреждения (ЛПУ), объединяющая в себе все предыдущие уровни иерархии. В других классификациях (например, С. А. Гаспаряна или Г. А. Хая) первый и второй уровень объединены в один – медицинские технологические системы. Настоящая работа связана с первым уровнем иерархической классификации и с перспективой развития до второго уровня. Современные системы диагностики и поддержки принятия решений в медицине [7–13] преимущественно рассматривают объект диагностики односторонне (гомогенно), решая узконаправленные задачи, применяющие ранее выработанные методы моделирования, как правило, в рамках одного из подходов искусственного интеллекта. В данной работе рассмотрены особенности и принципы диагностического процесса в стационаре (гетерогенная диагностика) и ЛПУ широкого профиля (гомогенная диагностика) на примере артериальной гипертензии (АГ), а также возможность повышения качества медицинских решений в ЛПУ широкого профиля за счет консультаций с моделью [14], имитирующей в компьютере коллективную работу врачей для поддержки принятия решений в процессе диагностики артериальной гипертензии, посредством инструментальной среды «Виртуальный консилиум».

2. О применении опыта и знаний гетерогенной диагностики артериальной гипертензии стационара в лечебно-профилактических учреждениях широкого профиля

Медицинская диагностика (от др.-греч. διαγνωστικός, «способный распознавать») [15] – раздел клинической медицины, изучающий содержание, методы и последовательные ступени процесса распознавания болезней или особых физиологических состояний. В узком смысле – это процесс распознавания болезни и оценки индивидуальных биологических особенностей и социального статуса субъекта, включающий целенаправленное медицинское обследование, истолкование полученных результатов и их обобщение в виде установленной диагноза. Используя результаты диагностики, врач снижает неопределенность о состоянии больного, определяет симптомы, классифицирует состояние. Возникает задача принятия индивидуального решения в условиях разнообразия перерабатываемой врачом информации об организме человека, а также неполноты, неточности, недоопределенности анатомических, физиологических и патологических характеристик пациента.

С.П. Боткин отличал «диагноз болезни» и «диагноз больного». Он сыграл большую роль в утверждении «диагноза больного» в клинической медицине. В своей клинической лекции в 1862 г. [16] говорил о том, что если диагностика болезни осуществляется на основании определения патологических изменений того или другого органа, то диагностика больного основывается на выявлении состояния всех остальных органов. Однако в XX веке А.С. Попов и В.Г. Кондратьев [17] замечают, что

противопоставление диагноза болезни диагнозу больного является не диалектическим, поскольку в современных условиях понятие «нозологическая форма» или «нозологическая единица» имеет не только семиологическое выражение, но и анатомическое, функциональное и этиологическое содержание. Здесь «нозологическая форма» или «нозологическая единица» [18] – определенная болезнь, которую выделяют в качестве самостоятельной на основе установленных причин, механизмов развития и характерных клинико-анатомических проявлений, отражающих преимущественное поражение тех или иных органов и систем организма; единица номенклатуры и классификации болезней. Противопоставление диагноза болезни диагнозу больного бесосновательно, лишено здравого смысла, поскольку индивидуальный диагноз – это по сути диагноз болезни у данного конкретного больного.

Позиция А.С. Попова и В.Г. Кондратьева соответствует клиническому эпикризу, сформулированному в истории болезни – медицинской карте стационарного больного (гетерогенная диагностика) – и содержащему сведения о природе, сущности, основных проявлениях и течении болезни. Однако в лечебно-профилактических учреждениях широкого профиля амбулаторно-поликлинического типа: фельдшерско-акушерские пункты, амбулатории, медико-санитарные части, поликлиники – чаще ставится диагноз болезни (гомогенная диагностика), сформулированный в соответствии с номенклатурой и классификацией болезни.

У 70% пациентов ЛПУ широкого профиля диагноз АГ «недовыявлен» в основном из-за инертности врачей по причине сомнений в степени риска, связанного с высоким артериальным давлением, и инертности врачей при нормальном артериальном давлении на первом приеме. Кроме того снижению качества диагностики в ЛПУ широкого профиля способствуют разнообразие в охватываемых разделах медицины и в типах симптоматической информации, а также её растущие объемы. В то время как в стационарных медицинских учреждениях (рис. 1) практически в 100% случаев имеет место, как адекватное проведение, так и отображение в медицинских картах данных анамнеза и физикального обследования с подтверждением диагноза лабораторно-инструментальными методами исследования.

В стационаре (на примере Калининградской клинической областной больницы) формируется полный дифференциальный диагноз АГ с привлечением до тринадцати врачей-экспертов (на рис. 1 – $M = \overline{I, I\bar{3}}$) – носителей знаний из различных разделов медицины: невролог, нефролог, сосудистый хирург, уролог, психолог, педиатр, акушер-гинеколог, онколог, окулист, врачи функциональной диагностики, эндокринолог, терапевт, кардиолог – лицо, принимающее решение (ЛПР). Они сотрудничают и дополняют друг друга, рассматривая организм пациента с различных точек зрения, каждый относительно своего опыта и теоретической подготовки.

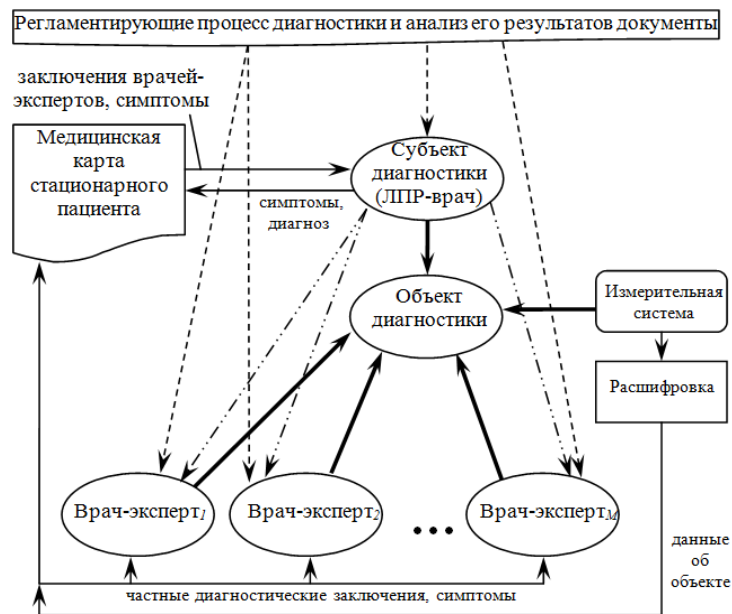


Рис. 1. – Концептуальная модель диагностического процесса в стационаре больницы

Обозначения: $- \rightarrow -$ – информационно-управляющие отношения «внешняя среда (регламентирующие документы) – субъект диагностики» и «внешняя среда – врач-эксперт»; $- \cdots \rightarrow$ – отношения сотрудничества «субъект диагностики – врач-эксперт»;

\rightarrow – информационно-управляющие отношения «врач-эксперт – медицинская карта», «медицинская карта – врач-эксперт», «субъект диагностики – медицинская карта», «медицинская карта – субъект диагностики», «измерительная система – медицинская карта»; \Rightarrow – отношения формирования множества значений показателей состояния здоровья объекта диагностики

Каждый объект (пациент) характеризуется множеством возможных значений показателей состояния – симптомами (при диагностике АГ выделяют порядка 800 значений показателей состояния здоровья пациента). При сборе анамнеза (опрос пациента) и физикальном обследовании (прослушивание, простукивание и прощупывание) пациента информация по симптомам заносится в медицинскую карту субъектами диагностики, врачами-экспертами узкой специализации, каждый из которых формирует промежуточные заключения о состоянии объекта диагностики в своей области медицинских знаний. При лабораторно-инструментальном обследовании множество симптомов пополняется измерительной системой после расшифровки врачами функциональной диагностики. Полученные исходные данные об объекте диагностики разнородны: измерения дают количественные и визуально-графические параметры (детерминированные переменные). От ЛПР (врач-кардиолог, вырабатывающий диагноз) и врачей-экспертов данные поступают в формате количественных, лингвистических четких и нечетких переменных. ЛПР изучает в медицинской карте симптомы и частные диагностические мнения врачей-экспертов и ставит заключительный диагноз. Врачам-экспертам также доступны симптомы и мнения других врачей-экспертов в медицинской карте. ЛПР и врачи-эксперты обследуют пациента и формируют диагностические заключения согласно нормативным документам, например [19].

Предлагается распространить опыт и знания стационарной диагностики АГ на диагностику в ЛПУ широкого профиля (рис. 2а) за счет внедрения информационной технологии «Виртуальный консилиум» (рис. 2б).

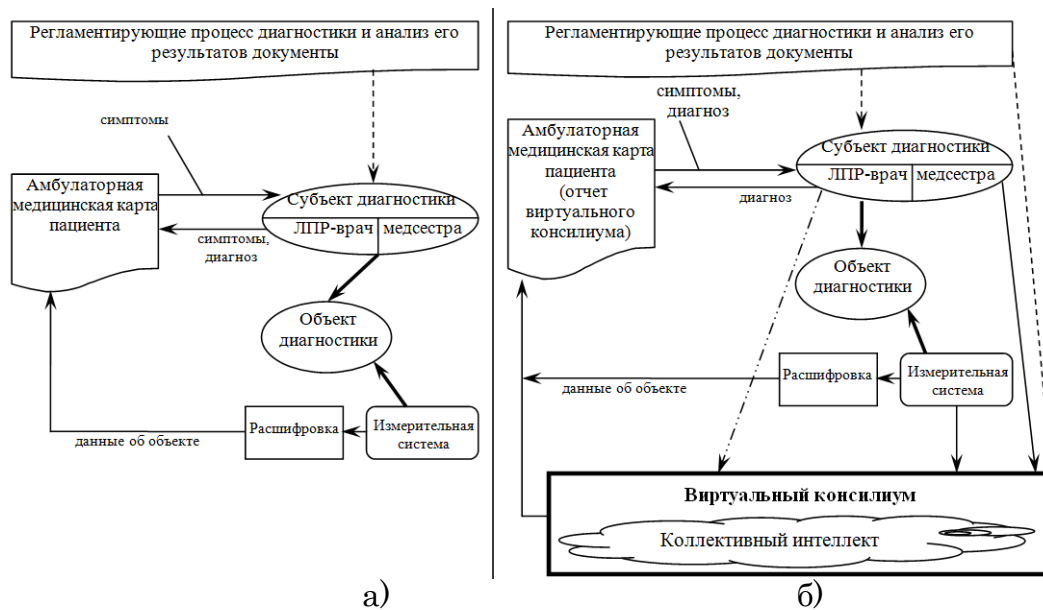


Рис. 2. – Концептуальная модель диагностического процесса в лечебно-профилактическом учреждении широкого профиля: а) без «Виртуального консилиума»; б) с применением «Виртуального консилиума»

Обозначения: а): $--\blacktriangleright$ – информационно-управляющие отношения «внешняя среда (регламентирующие документы) – субъект диагностики»; \longrightarrow – информационно-управляющие отношения «субъект диагностики – медицинская карта», «медицинская карта – субъект диагностики», «измерительная система – медицинская карта»; \blacktriangleright – отношения формирования множества значений показателей состояния здоровья объекта диагностики;

б): $--\blacktriangleright$ – информационно-управляющие отношения «внешняя среда – субъект диагностики» и «внешняя среда – виртуальный консилиум»; $- \cdot \cdot \blacktriangleright$ – отношения сотрудничества «субъект диагностики – виртуальный консилиум»; \longrightarrow – информационно-управляющие отношения «виртуальный консилиум – медицинская карта», «субъект диагностики – медицинская карта», «медицинская карта – субъект диагностики», «измерительная система – медицинская карта», «измерительная система – виртуальный консилиум»; \blacktriangleright – отношения формирования множества значений показателей состояния здоровья объекта диагностики;

В ЛПУ широкого профиля (рис. 2а) субъект диагностики – это врач общей практики или терапевт (иногда кардиолог, но зачастую без опыта работы, к которому направляет терапевт сразу же при выявлении повышенного артериального давления), это врач «прагматик-фактолог», редко задумывающийся над тем, что делается «вокруг» и какое значение могут иметь добытые им факты для понимания работы организма в целом, а также объединяющий в себе роли ЛПП-врача и экспертов-врачей узкой специализации. Субъект диагностики включает и медсестру, помогающую врачу на приеме пациентов.

Предлагаемая технология «Виртуальный консилиум» [14] и первая версия инструментальной среды для поддержки принятия диагностических решений по АГ [20], архитектура которой представлена на рис. 3, а структура в [14], ограничена работой со знаниями шести врачей-экспертов разной специализации, решающими двенадцать функциональных подзадач, возникающих в 90% случаев диагностики АГ, и также ограничивается пациентами старше 18, без особых состояний, нет распознавания снимков, не предусматривается назначение лечения и не диагностируется ряд симптоматических артериальных гипертензий.

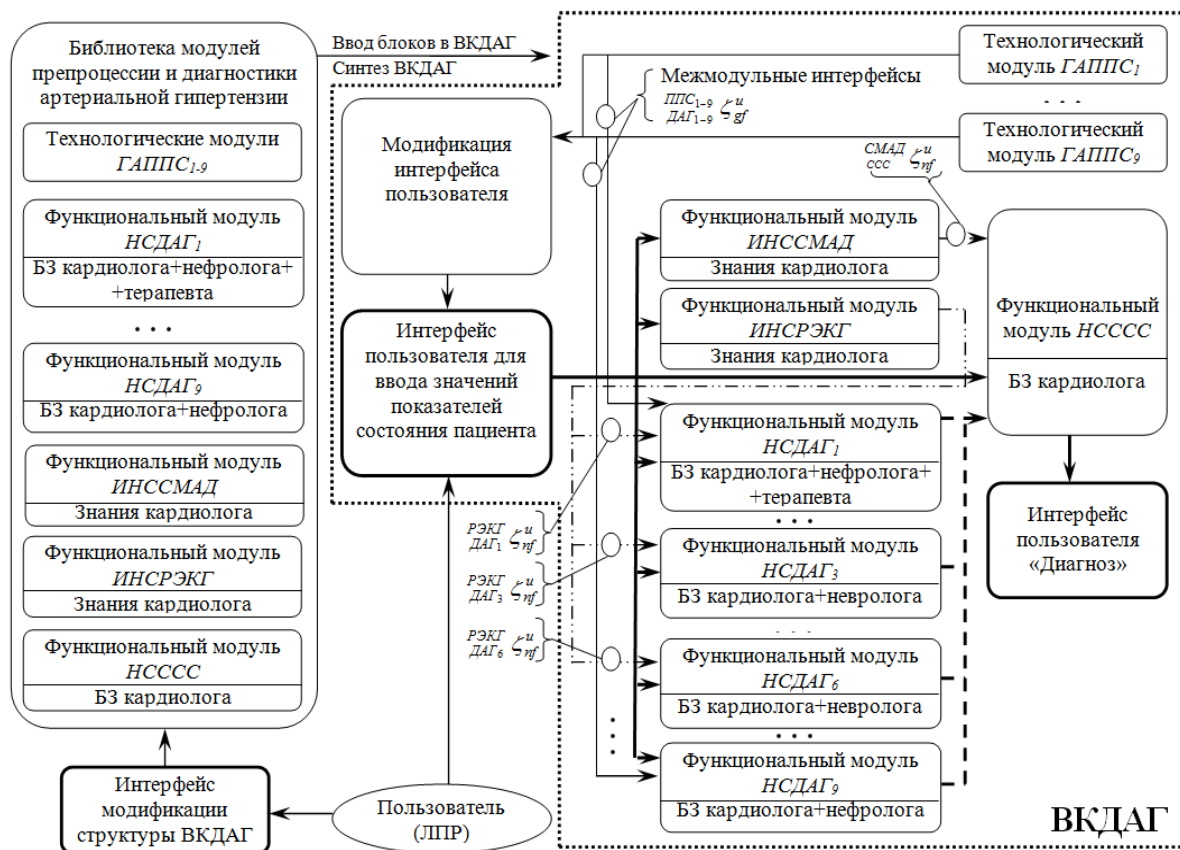


Рис. 3. – Архитектура Виртуального консилиума для диагностики артериальной гипертензии

В табл. ниже приведено описание функций, выполняемых всеми блоками модулей Виртуального консилиума для диагностики артериальной гипертензии (VKDAГ).

Таблица – Описание блоков архитектуры Виртуального консилиума для диагностики артериальной гипертензии

Наименование блока	Функции
1	2
Технологический модуль i -ый	Построение информативного набора признаков при диагностике заболеваний (ППС, по умолчанию включен) – минимизирует общее число входных переменных i -ого функционального модуля (реализованы посредством методологии генетических алгоритмов)
Функциональный модуль i -ый ($HСДАГ_i$), включающий базу знаний врача(ей)	Классификация состояния здоровья пациента в рамках i -ой подзадачи из декомпозиции сложной задачи диагностики АГ [14], выбирается для включения в «Виртуальный консилиум» (по умолчанию включен) (реализованы посредством методологии нечетких систем)
Функциональный модуль $ИНСРЭКГ$, моделирующий знания кардиолога	Классификация патологического состояния пациента по его электрокардиограмме (реализован посредством методологии искусственных нейронных сетей)

1	2
Функциональный модуль <i>ИНССМАД</i> , моделирующий знания кардиолога	Прогноз нормальных значений суточного мониторинга артериального давления (СМАД) и вычисление отклонения от них данных СМАД пациента (реализован посредством методологии искусственных нейронных сетей)
Функциональный модуль <i>НСССС</i> , включающий базу знаний кардиолога	Формирование заключительного диагноза артериальной гипертензии (всегда в составе Виртуального консилиума, реализован посредством методологии нечетких систем)

Функции, вход и выход всех блоков, а также интерфейсы Виртуального консилиума для диагностики АГ и библиотека модулей подробно описаны в [20].

Архитектура ВКДАГ включает межмодульные интерфейсы для модулей, реализованных посредством различных методологий гибридных интеллектуальных систем:

ζ_{gf}^{u} между технологическими модулями – генетические алгоритмы (g) и функциональными модулями $НСДАГ_{1-9}$ – нечеткие системы (f); ζ_{nf}^{u} , ζ_{nf}^{u} , ζ_{nf}^{u} между функциональным модулем *ИНСПЭКГ* – искусственные нейронные сети (n) и функциональными модулями $НСДАГ_1$, $НСДАГ_3$, $НСДАГ_6$ – нечеткие системы (f) соответственно; ζ_{nf}^{u} между функциональным модулем *ИНССМАД* – искусственные нейронные сети (n) и функциональным модулем *НСССС* – нечеткие системы (f).

Консилиум (рис. 3) запускает интерфейс пользователя, ЛПР-врача — «Интерфейс модификации структуры ВКДАГ», посредством которого включаются функциональные и технологические модули в работу системы ВКДАГ: модуль «Анализ СМАД», модуль «Распознавание ЭКГ», модули подзадач из группы «Построение информативного набора признаков (симптомов) при диагностике заболеваний» и модули подзадач из группы «Диагностика критериев оценки сердечно-сосудистого риска и вторичной АГ у пациента». Все выбранные i -ые технологические модули запускаются, решают технологические подзадачи из группы «Построение информативного набора признаков (симптомов) при диагностике заболеваний» и передают информацию на блок «Модификация интерфейса пользователя». Он деактивирует показатели состояния здоровья на «Интерфейсе пользователя для ввода значений показателей состояния здоровья пациента» и корректирует работу i -ого функционального модуля подзадач $ДАГ_1, \dots, ДАГ_9$: диагностики поражений органов мишеней, факторов риска, цереброваскулярных болезней, метаболического синдрома и сахарного диабета, заболеваний периферических артерий, ишемической болезни сердца, эндокринной АГ, паренхиматозной нефропатии и реноваскулярной АГ соответственно. Далее активируется «Интерфейс пользователя для ввода значений показателей состояния здоровья пациента», посредством которого вводятся значения показателей состояния здоровья пациента (симптомы), передаваемые функциональным модулям $НСДАГ_1, \dots, НСДАГ_9$, решающим подзадачи $ДАГ_1, \dots, ДАГ_9$. Последние в свою очередь передают информацию о патологиях, выявленных ими у пациента, функциональному модулю *НСССС*, решающему подзадачу «Оценка степени и стадии артериальной гипертензии, степени риска сердечно-сосудистых заболеваний», и являются результатами моделирования принятия решения экспертами, врачами смежных специальностей — кардиологом (в том числе и в качестве ЛПР), неврологом, нефрологом, терапевтом, эндокринологом, урологом. Модуль *НСССС* моделирует принятие решения ЛПР — врачом-кардиологом.

В библиотеке ВКДАГ содержится функциональный модуль *ИНСПЭКГ* (включается в работу ВКДАГ посредством «Интерфейса модификации структуры ВКДАГ»), пе-

редающий информацию (см. табл. выше) на модули диагностики поражений органов мишеней (на рис. 3 — это *НСДАГ₁*), цереброваскулярных болезней (на рис. 3 — это *НСДАГ₃*) и ишемической болезни сердца (на рис. 3 — это *НСДАГ₆*). Помимо этого библиотека Виртуального консилиума для диагностики артериальной гипертензии содержит функциональный модуль *ИНССМАД* (включается в работу ВКДАГ посредством «*Интерфейса модификации структуры ВКДАГ*»), передающий информацию о требуемых нормальных значениях суточного артериального давления на функциональный модуль *НСССС*.

При лабораторной апробации [14] прототипа «Виртуального консилиума» вычислялась среднеквадратическая ошибка постановки диагноза. В итоге диагноз прототипа верен в 84% случаев. Кроме этого, исследовалось время диагностики. Оно складывается из трех составляющих: времени на собственно диагностику и занесения информации в карту, времени обработки информации и времени записи заключения. Сумму последних двух составляющих с применением прототипа виртуального консилиума сократилось, примерно, до 30 сек.

Предварительные результаты лабораторных экспериментов с прототипом дали обнадеживающие результаты и позволяют предположить, что внедрение «Виртуального консилиума» в ЛПУ широкого профиля повысит эффективность и качество индивидуальных диагностических решений ЛПР-врача до уровня стационара, поскольку консультации будут получены у искусственного коллективного интеллекта обладающего синергией. При этом ЛПР-врач (рис. 2б) обследует пациента, а медсестра заносит симптомы в компьютер. «Виртуальный консилиум» анализирует полученную информацию и выводит заключительный диагноз в форме аналогичной выписке из истории болезни стационара. ЛПР-врач при несогласии с диагнозом может повторно обратиться к «Виртуальному консилиуму».

3. Заключение

Развитие современного общества определяет успешное и бесперебойное функционирование его основных ресурсов: человеческих и материально-технических. В этой связи особенно актуальна дифференциальная диагностика состояния здоровья нации и исправности технических объектов и систем. Диагностические решения принимаются в условиях ограниченности ресурсов, недостатка времени, дефицита экспертов, неопределенности информации об окружающем мире и исследуемом объекте. Количество и разнообразие перерабатываемой при принятии решения информации увеличивается невиданными темпами. Для снижения рисков неверных диагностических решений перспективна автоматизация обслуживания диагностического процесса и создание программно-аппаратных средств. При этом объекты диагностики, в частности, пациенты в медицине рассматриваются, как гетерогенные системы.

Сейчас врачи и системы поддержки принятия диагностических решений преимущественно «видят» пациента односторонне, одноаспектно, с позиций одной профессии, одной дисциплины, одной школы, что характерно для гомогенных систем и ведет к решению задач диагностики не системно, когда разнообразные знания не взаимодействуют, не сотрудничают для достижения общей цели, что по определению не создает условий для возникновения синергетического, системного эффекта. В основе же «Виртуального консилиума», работа над которым выполняется при поддержке гранта РФФИ № 16-07-00272 А, лежит использование взаимодействующих друг с другом моделей разнородных знаний, способных имитировать коллективные решения (консилиумы), разнообразие хранимой и перерабатываемой информации в которых релевантно разнообразию информации в диагностических ситуациях. Такие системы потенциально способны динамично синтезировать над неоднородным модельным полем новый метод выра-

ботки диагностического заключения для каждого объекта диагностики индивидуально и применять его для подготовки рекомендаций лицу, принимающему решения.

Литература

1. Fisher P.R., Kurlander D.T. Microcomputers in medical diagnosis // Proc. Ann. Conf. Nashville, Tenn., Oct. 27 – 29, 1980, New York, p. 75– 79.
2. Гублер Е.В. Вычислительные методы анализа и распознавания патологических процессов. – Л.: Медицина, 1978. – 293 с.
3. Мисюк Н.С., Семак А.Е., Довнар И.Н. и др. Результаты математического прогнозирования мозговых инсультов. – В кн.: Вопросы ранней диагностики и лечения нервных и психологических заболеваний. – Каунас. – 1979. – с. 97–98.
4. Амосов Н.М., Минцер О.П., Палец Б.Л. О возможностях кибернетики в медицине. – Кардиология. – 1977. – №7. – С. 19–25.
5. Landucci L., Macorata A., Marchesi A. et. al. Real time computed based: electrocardiographic and hemodynamic monitoring in CCU. – In: Med. Inform. Berlin, 1979. Proc. Int. conf. Med. Comput., Berlin, 1979. Berlin e.a., 1979, p. 325–329.
6. Кобринский Б.А., Зарубина Т.В. Медицинская информатика. 4-е изд., перераб и доп. – М. : Издательский центр «Академия», 2013. – 192 с.
7. Чеснокова И. В. Оптимизация лечебно-диагностического процесса у больных артериальной гипертензией на догоспитальном этапе: Автореф. дисс. ... докт. мед. наук. Воронеж: Воронежская гос. мед. акад. им. Н. Н. Бурденко, 2009. – С. 46.
8. Шулакова М. А. Информационная поддержка принятия решений при диагностике артериальной гипертонии на основе методов гибридного интеллекта: Автореф. дисс. ... канд. техн. наук. Воронеж, Воронежский гос. техн. ун-т, 2012. – С. 16.
9. Аль Мабрук Мохаммад. Аппаратно—программные средства и алгоритмы распознавания патологий сердца на основе перцептронных сетей: Дисс. ... канд. техн. наук. — Владимир: Рязан. гос. радиотехн. ун-т., 2011. – 203 с.
10. Реброва О. Ю. Математические алгоритмы и экспертные системы в дифференциальной диагностике инсультов: Дисс. ... докт. мед. наук. М.: НИИ неврологии РАМН, 2003. – С. 325.
11. Automated System For Diagnosis Of Major Coronary Artery Stenosis. <http://www.sciencedaily.com/releases/2009/04/090423132609.htm>
12. Бессонова Т. В. Методы интеллектуальной поддержки принятия решений в задачах диагностики и лечения хронической сердечной недостаточности : диссертация ... кандидата технических наук : 05.13.01 / Бессонова Татьяна Владимировна; [Место защиты: Воронеж. гос. техн. ун-т]. - Воронеж, 2008. – 135 с.
13. Грахов А. А. Автоматизированная система прогнозирования, ранней и дифференциальной диагностики ишемической болезни сердца на основе нечетких сетевых моделей : технические и медицинские системы : диссертация ... кандидата технических наук : 05.13.01 / Грахов Алексей Алексеевич; [Место защиты: Кур. гос. техн. ун-т].- Курск, 2008. – 154 с.
14. Кириков И.А., Колесников А.В., Румовская С.Б. Исследование лабораторного прототипа искусственной гетерогенной системы для диагностики артериальной гипертензии // Системы и средства информатики, 2014. Т.24. №.3. С. 121–144.
15. Малая медицинская энциклопедия в 6-ти томах. Т 2: Грудь – Куммеля болезнь / Под ред. Покровского В.И. – М.: «Медицина», 1991 г. С. 89.
16. Боткин С.П. Из первой клинической лекции // Медицинский вестник, 1862. No.41. С. 392.

17. Попов А. С., Кондратьев В. Г. Очерки методологии клинического мышления. Л., Медицина, 1972. – 182 с.
18. Малая медицинская энциклопедия в 6-ти томах. Т 4: Нефротомия – Почечная недостаточность / Под ред. Покровского В.И. – М.: «Медицина», 1996 г. – С. 17.
19. Комитет экспертов ВНОК. Диагностика и лечение артериальной гипертензии. Российские рекомендации // Системные гипертензии, 2010. Вып. 3. М.: С. 5–26.
20. Кириков И.А., Колесников А.В., Румовская С.Б. Функциональная гибридная интеллектуальная система для поддержки принятия решений при диагностике артериальной гипертензии // Системы и средства информатики, 2014. Т.24. No.1. С. 153–179.
21. Кириков И.А., Колесников А.В., Румовская С.Б. Исследование сложной задачи диагностики артериальной гипертензии в методологии искусственных гетерогенных систем // Научный журнал «Системы и средства информатики», 2013. Т. 23. No. 2. С. 96 – 114.

СЕКЦИЯ «ПАРАЛЛЕЛЬНЫЕ И РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ»

СРЕДА КОМПЛЕКСНОГО АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМОВ БАЛАНСИРОВКИ В ПАРАЛЛЕЛЬНОМ МЕТОДЕ ВЕТВЕЙ И ГРАНИЦ⁵

Орлов Ю.В.

Вычислительный центр имени А. А. Дородницына ФИЦ ИУ РАН
119333, Москва, ул. Вавилова, 40, justice1786@gmail.com

Аннотация

В работе описывается среда комплексного анализа производительности алгоритмов балансировки вычислительной нагрузки в параллельном методе ветвей и границ. Основное назначение разработанной среды – поддержка исследования производительности алгоритмов балансировки и выяснение причин ее снижения. Рассмотрены основные подходы к визуализации производительности, продемонстрированы их достоинства и недостатки. Приведена общая схема и принципы работы разработанного программного инструментария.

1. Введение

Многие задачи глобальной оптимизации относятся к классу NP и их решение требует значительных вычислительных ресурсов. Поэтому представляется целесообразным применение методов параллельных [1,2] и распределенных вычислений [3]. Метод ветвей и границ (МВГ) является одним из наиболее распространенных алгоритмов решения задач дискретной оптимизации. В его основе лежит идея декомпозиции, которая делает естественным применение параллельных вычислений. Этим фактом можно объяснить значительный интерес исследователей к вопросам параллельной реализации МВГ. Обзор различных подходов можно найти в работах [4-6]. Основной проблемой при параллельной реализации методов типа ветвей и границ является адекватная балансировка вычислительной нагрузки между параллельными процессорами. Так как информационный граф алгоритма [7], в данном случае представляющий собой дерево, заранее не известен, особую важность получают методы динамической балансировки, перераспределяющие вычисления в процессе работы в зависимости от загрузки процессоров.

В ближайшее время ожидается, что суперкомпьютеры обретут производительность порядка эксафлопс (около 10^{18} операций с плавающей точкой в секунду). Основным способом увеличения производительности является наращивание количества ядер, число которых в некоторых современных системах [8] уже превосходит 10^{16} . В такой ситуации балансировка нагрузки становится весьма нетривиальной задачей. Следовательно, необходим развитый инструментарий для анализа производительности алгоритмов балансировки.

Одной из важнейших задач комплексного анализа производительности таких алгоритмов является выявление причин потерь производительности. Часто без

⁵ Работа выполнена при поддержке РФФИ (проекты № 16-07-00873 А и № 16-07-00458 А).

визуализации, используя одну лишь трассу, сделать это практически невозможно, так как трассы выполнения параллельных программ обычно очень велики по объему и слабо поддаются визуальному анализу.

Для наиболее полного анализа производительности параллельных алгоритмов оптимизации необходимо иметь возможность:

- осуществлять визуализацию работы многопроцессорной системы в динамике;
- осуществлять визуализацию коммуникаций между процессами;
- вычислять ускорение и эффективность системы в целом.

На сегодняшний день уже существуют визуальные программные комплексы, позволяющие анализировать производительность процессоров и ядер на современной вычислительной технике. Однако данные программные продукты не позволяют осуществлять визуализацию низкоуровневых коммуникаций между процессами. Поэтому, проблема разработки и реализации визуальной среды для анализа производительности параллельных алгоритмов оптимизации остается актуальной. Также практика показала, что реализация того или иного функционала среды требует научного подхода. Не без внимания остаются и аспекты дальнейшего развития среды комплексного анализа, о которых будет сказано ниже.

2. Основные принципы работы среды

Важнейшей составляющей разработки алгоритмов управления распределением вычислений в многопроцессорной системе является отладка, позволяющая выявить слабые стороны тестируемого алгоритма. Отладка на реальной многопроцессорной системе дает наиболее полную информацию о производительности алгоритма. Но такая отладка весьма трудоемка, так как требует большого числа запусков параллельного приложения.

Для того, чтобы проводить тестирование алгоритмов балансировки без проведения ресурсоемких вычислений, был разработан симулятор многопроцессорной системы BNB-Simulator[9] на базе компонентов библиотеки BNB-Solver[10,11]. Данная библиотека предлагает набор модулей для разработки параллельных приложений. Симулятор имитирует выполнение реального параллельного приложения, а его трасса имеет тот же формат, что и трасса параллельного приложения, разработанного на основе BNB-Solver. Таким образом, симулятор позволяет отлаживать алгоритм, заложенный в программе, который затем можно использовать как отдельный модуль параллельного приложения, запускаемого на многопроцессорной системе.

Симулятор представляет собой приложение, управляемое через файл настроек settings.json. Это в свою очередь позволяет непосредственно через графический интерфейс визуальной среды:

- запускать симулятор, передавая ему параметры и получая от него в ответ трассу в виде потока байт;
- изменять настройки симулятора.

Также среда комплексного анализа позволяет загружать трассу из файла с расширением «.trc». Таким образом, визуальная среда позволяет работать с трассой установленного формата, независимо от источника ее получения (реального приложения или симулятора).

Изначально предполагалось, что среда комплексного анализа в процессе своей работы будет хранить данные трассы внутри двумерного массива процессоров размерности $n*m$, где n – количество процессоров, а m – количество проработанного времени, измеряемого специальными метками. Однако, такой подход оказался неприемлемым при обработке трасс большого размера, так как при превышении лимита оперативной памяти программа начинает задействовать пространство жесткого

диска машины, что в свою очередь значительно замедляло ее работу, а в ряде случаев приводило к аварийному завершению программы.

Таким образом, понадобился более гибкий способ обработки трассы. В результате было принято решение делить в процессе обработки трассу на множество файлов, состоящих из результирующей и содержательной частей. Содержательная часть описывает работу подмножества процессоров на некотором промежутке времени. Результирующая часть содержит итоговые статистические данные о результате работы подмножества процессоров на установленном промежутке времени. Для обеспечения быстрого доступа к таким данным в процессе счета программы результирующая часть помещается в начало файла. Результаты экспериментального исследования нового алгоритма показали, что по мере увеличения размера трассы скорость ее обработки (парсинг данных, расчет статистики по ускорению и эффективности) падает незначительно по сравнению с алгоритмом, использованным ранее.

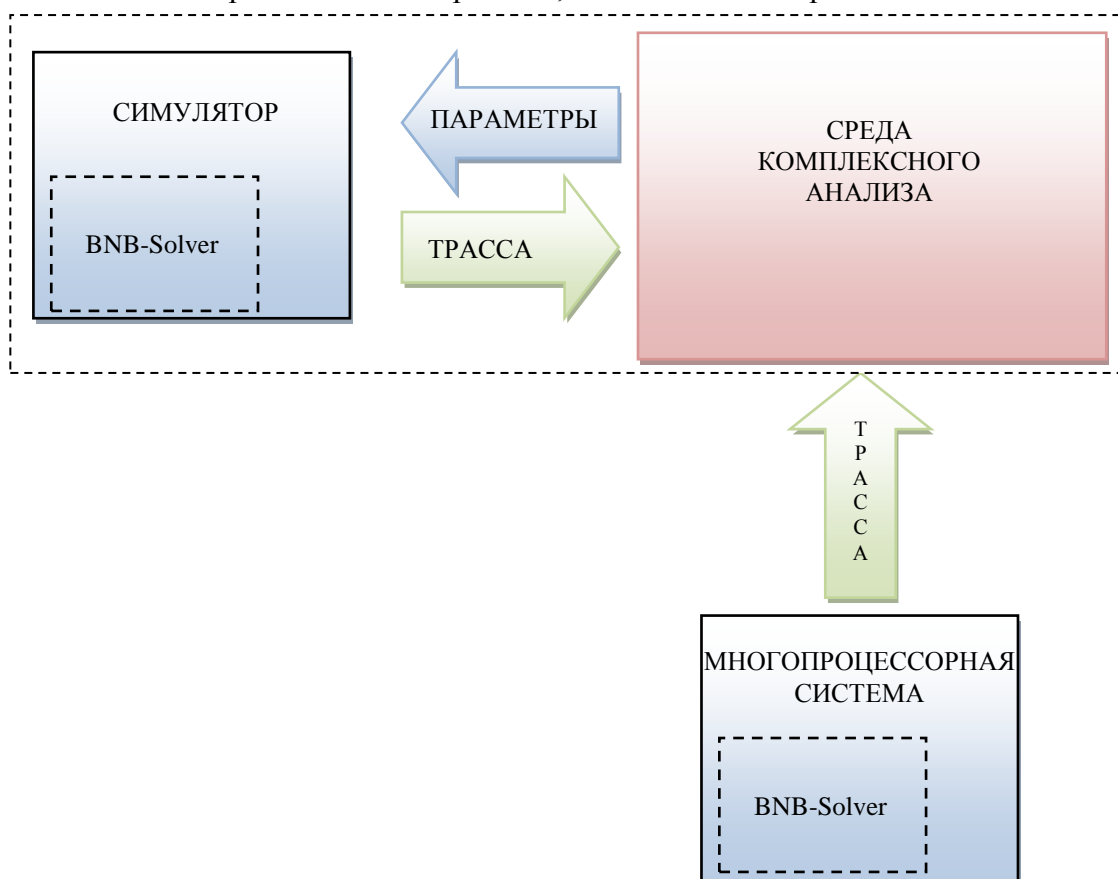


Рис. 1. – Общая схема работы визуальной среды

Трасса содержит информацию о низкоуровневых командах и событиях каждого из процессов, которые позволяют визуализировать передачи данных между процессорами. Ранее такая информация хранилась в виде списка объектов обмена данными. Однако, при работе с трассами большого размера можно было столкнуться со значительными замедлениями в работе среды. В результате было принято решение разделить информацию о передаче данных на части и хранить их в отдельных файлах. В названии этих файлов указан промежуток времени, в котором происходят передачи данных. Это позволило ускорить навигацию по трассе.

3. Основные компоненты визуальной среды

Разработанная среда предоставляет следующую функциональность:

- загрузка и разбор трассы;
- визуализация графиков работы процессоров;
- визуализация графиков таблицы процессоров;
- визуализация обмена данными между процессами;
- расчет и вывод статистики.

Окно компонентов управления (рис. 2) представляет собой интерфейс, напоминающий мультимедийный проигрыватель. Данное окно содержит полосу прокрутки, максимальное значение которой равно максимальному значению логического времени работы системы, кнопки для управления прокруткой («Rewind», «Play», «Pause», «Stop», «Forward») и слайдер для регулирования скорости воспроизведения действий процессоров. При изменении состояния полосы прокрутки обновляется открытое окно визуализации. Такой способ организации управления визуализацией позволяет абстрагироваться от времени работы многоядерной системы с целью наиболее детального исследования поведения каждого процессора и системы в целом.



Рис. 2. Окно компонентов управления

Для более детального исследования поведения процессов на определенном промежутке времени удобно использовать окно визуализации графиков производительности (рис. 3). В данном окне поведение каждого процесса представлено в виде набора графиков на координатной плоскости, отображающими состояние процесса в конкретный момент времени. Красным цветом отображается состояние ожидания, синим – состояние счета, зеленым – состояние отправки данных другому процессу.

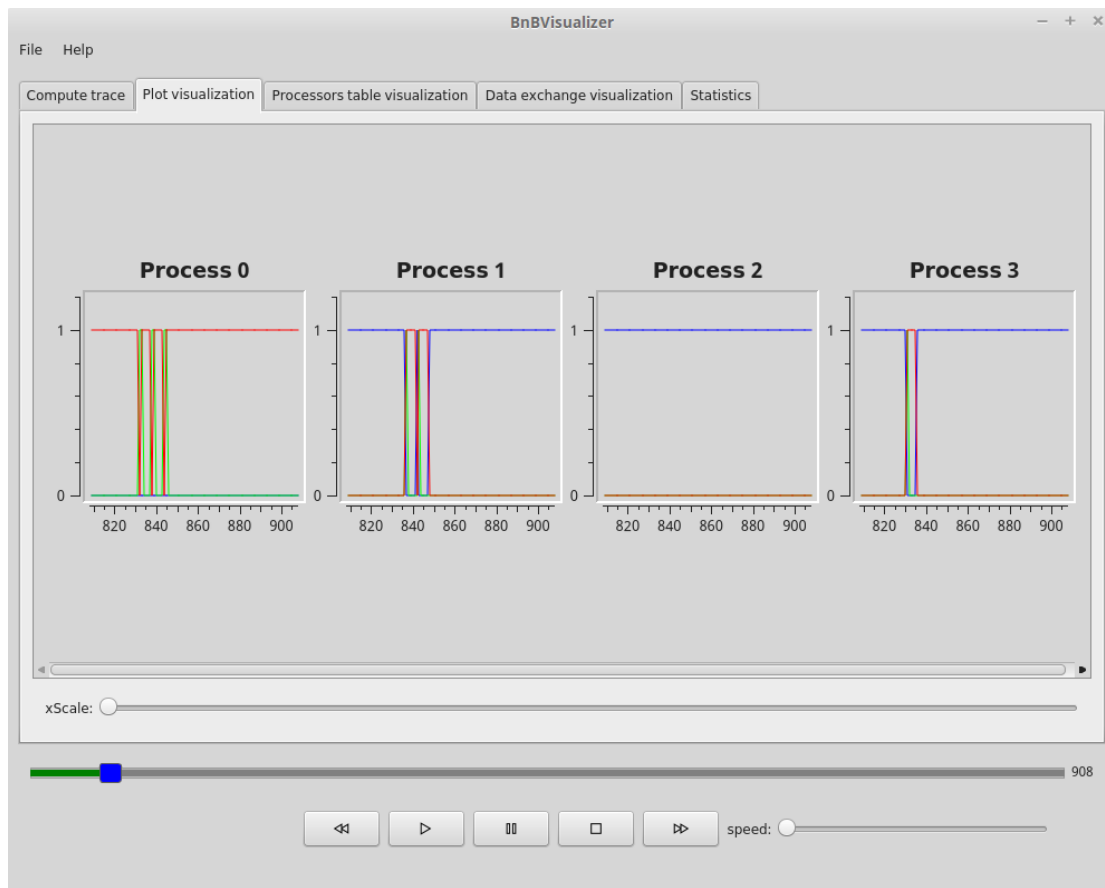


Рис. 3. Визуализация графиков загрузки процессоров

Такой подход удобен при небольшом (не более 10) числе процессоров. Если запуск параллельного приложения осуществляется на довольно большом количестве узлов, удобнее использовать окно визуализации таблицы процессоров (рис. 4). Каждый процесс в данном окне изображен в виде квадрата. В зависимости от своего состояния квадрат меняет цвет. Такой способ визуализации не позволяет нам исследовать поведение процесса на отрезке времени, зато дает возможность видеть состояние всей системы в каждый момент времени ее работы. Чтобы узнать номер процесса, соответствующий квадрату, достаточно навести на квадрат курсор мыши: номер отобразится во всплывающей подсказке.

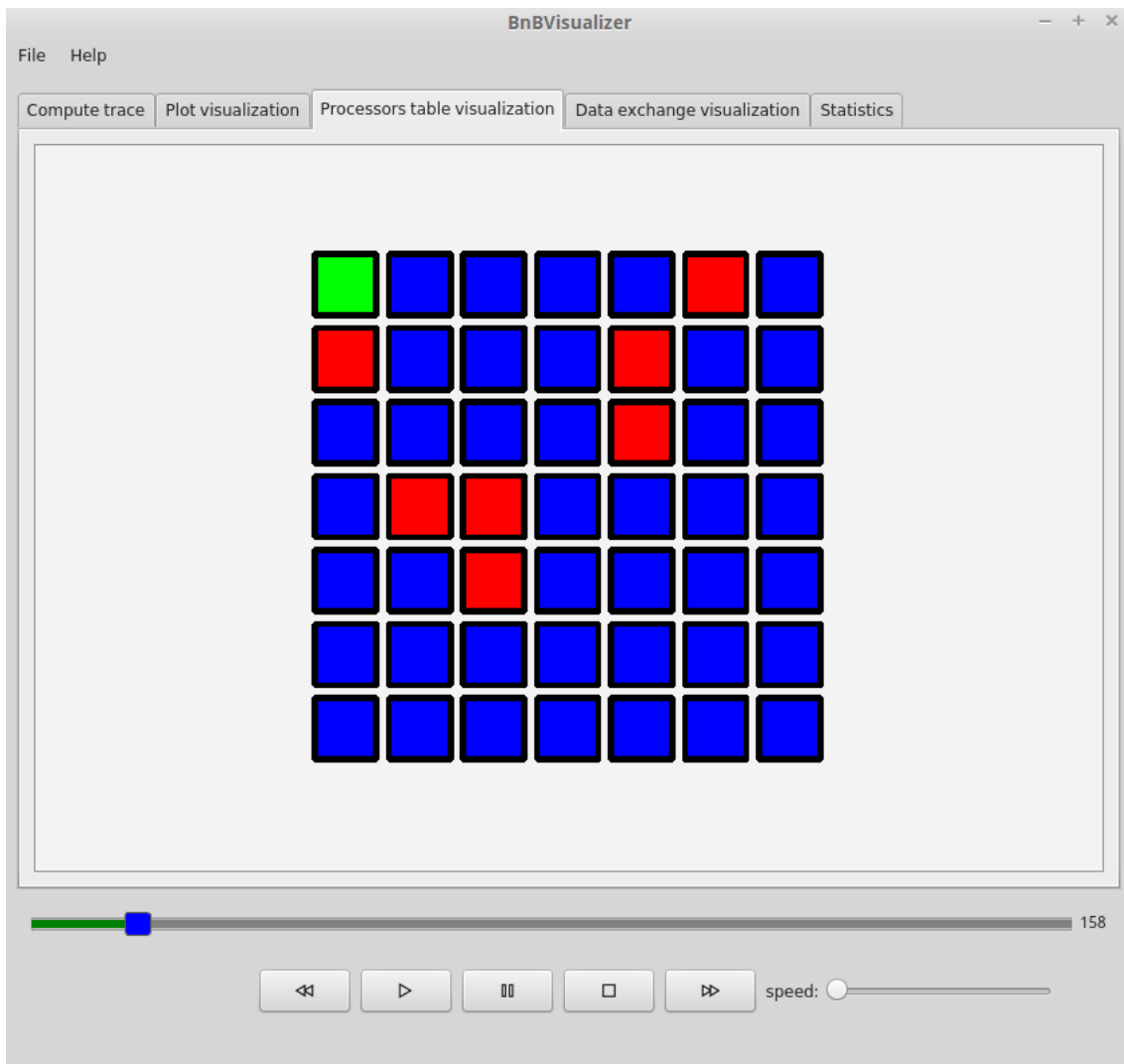


Рис.4. Визуализация таблицы загрузки процессоров

Для полноценного анализа параллельного алгоритма оптимизации одной лишь информации о состоянии процессов недостаточно. Часто при отладке алгоритма по трассе нужно обладать информацией о коммуникации процессов. С этой целью в среду комплексного анализа было добавлено окно визуализации обмена данными (рис. 5).

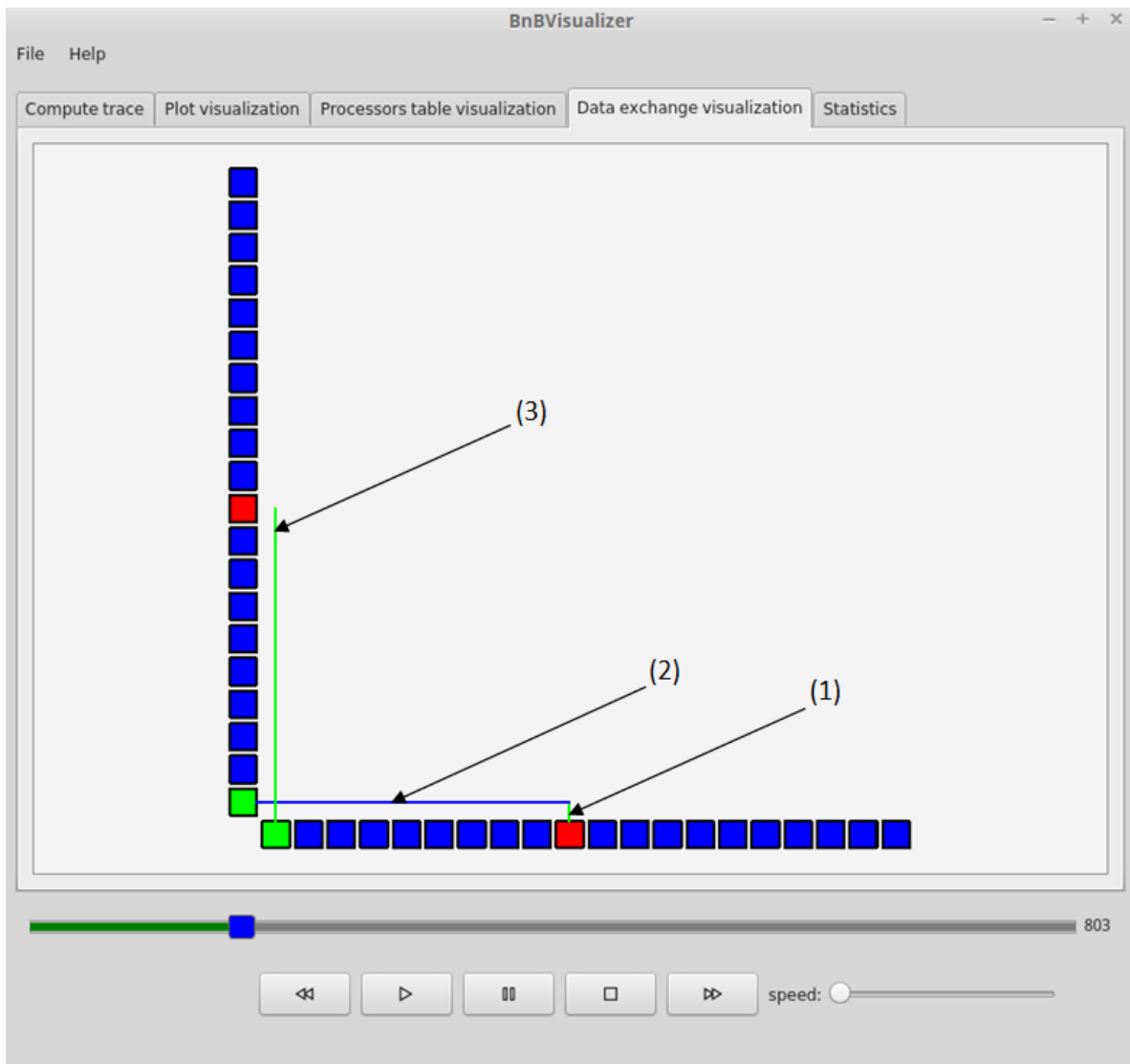


Рис. 5. Визуализация обмена данными. (1) – процесс с номером 9 отправил данные процессу с номером 0 и находится в состоянии ожидания; (2) – процесс с номером 0 получил отправленные ему процессом 9 данные; (3) – процесс с номером 0 отправил новые данные процессу 9

В данном окне каждому процессу соответствует два квадрата, меняющих цвет в зависимости от состояния процесса. Квадраты выстраиваются в два ряда, перпендикулярных по отношению друг к другу, от левого нижнего угла, что позволяет изображать процесс передачи сообщений при помощи цветных линий. Горизонтальный ряд соответствует процессорам-отправителям, а вертикальный – процессорам-получателям. Зеленая линия, исходящая вверх от отправителя, символизирует отправку сообщения, синяя линия, исходящая вправо от получателя, обозначает получение сообщения.

Такая визуализация дает возможность выявлять логические ошибки в алгоритме балансировки нагрузки. Так, например, можно выявлять тупиковые ситуации по трассе, используя простейший алгоритм, приведенный на рисунке 6. Если все процессы находятся в ожидании на последней метке времени, и при этом не зафиксировано ни одной передачи между ними, можно сделать вывод, что один или несколько процессов

захватили ресурсы и по какой-то причине не передали их другим процессам, ожидающим данные ресурсы.

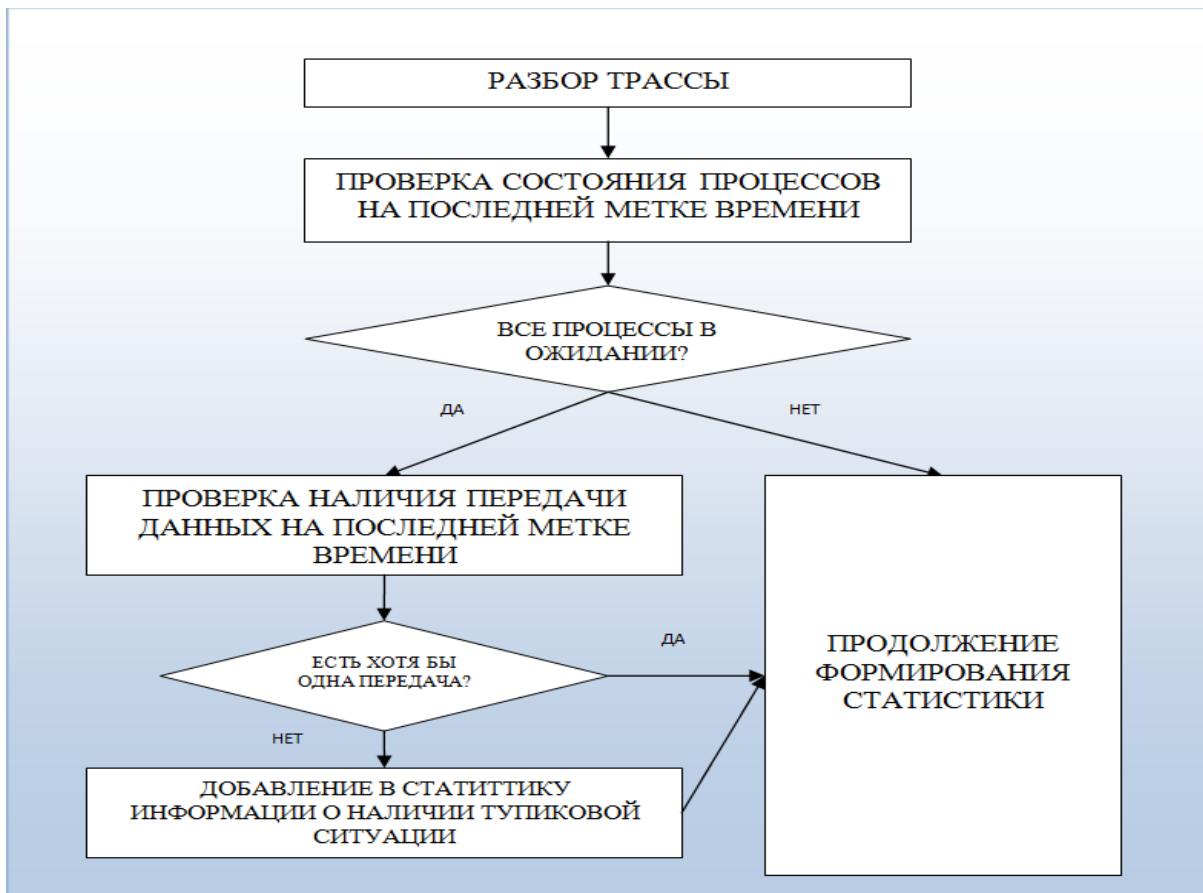


Рис. 6. Схема алгоритма обнаружения тупиковых ситуаций

На рисунке 7 (справа) видно, как процесс с номером 3 (начиная с 0) находится в состоянии ожидания, отправив перед этим данные процессу с номером 0. Процесс с номером 0 сразу же получает от него результаты и направляет новую порцию данных для счета. На том же рисунке (слева) видна ситуация, когда все три рабочих процесса завершили счет и перешли в состояние ожидания, не отправив перед этим результаты главному процессу, ожидающему эти данные.

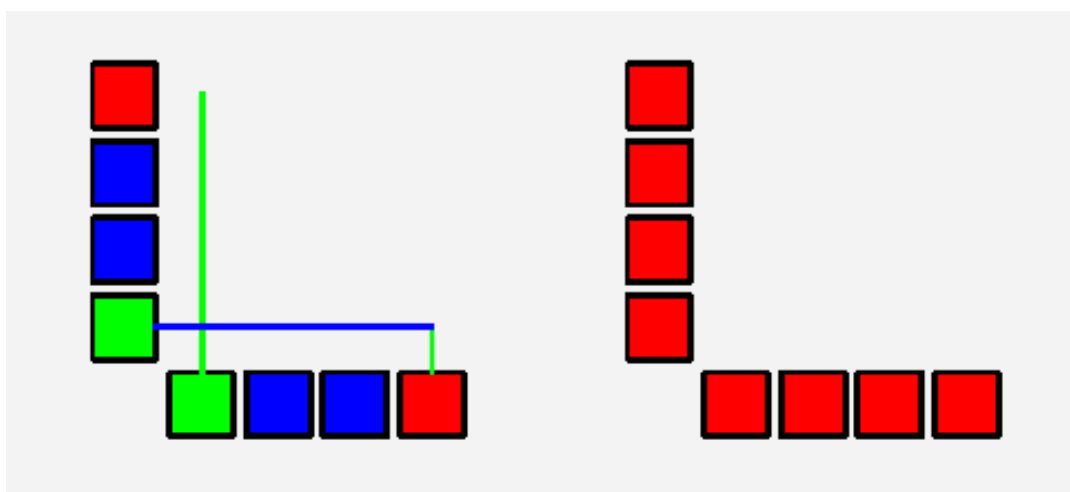


Рис. 7. Демонстрация обмена данными между процессами (слева) и отсутствия такого обмена между процессами, находящимися в ожидании (справа)

Для проведения сравнительного анализа работы параллельных программ на основе различных алгоритмов балансировки важными показателями являются такие метрики, как параллельная эффективность E и параллельное ускорение S [12].

$$S = \frac{t_{\text{послед}}}{t_{\text{пар}}}, \quad E = \frac{S}{n}.$$

Среда комплексного анализа автоматически формирует статистику для всей системы и по каждому отдельному процессу. Статистика состоит из графиков соотношения пиковой и реальной производительности, информации о выявленных логических ошибках (выводится, если среде комплексного анализа удастся автоматически их обнаружить), информации об ускорении и эффективности вычислительной системы и информации по каждому из процессов (рис.8).

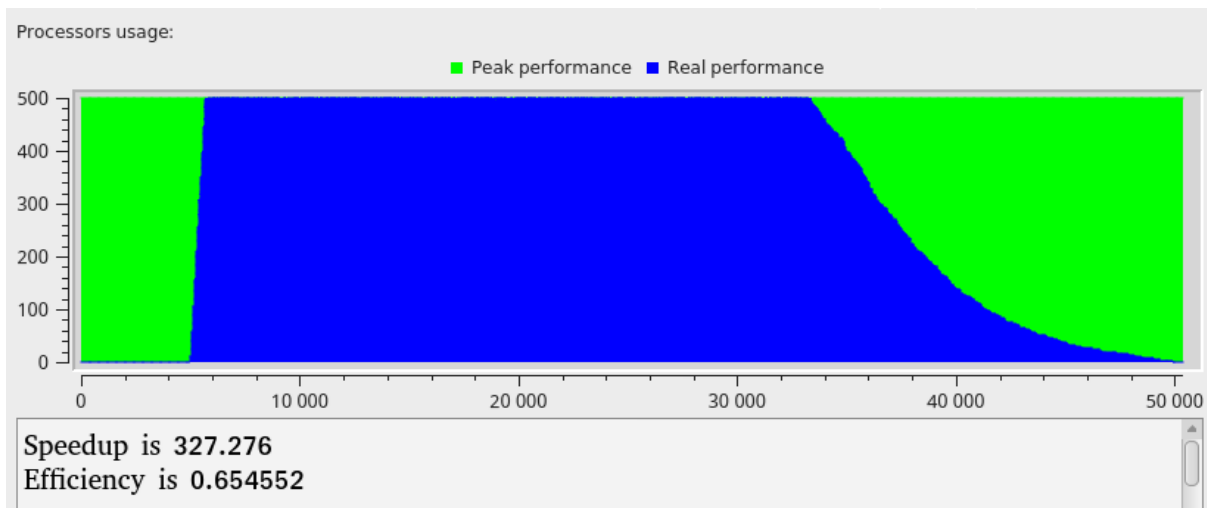


Рис. 8. Статистика по результатам работы вычислительной системы

4. Дальнейшее развитие среды

Функционал среды комплексного анализа нуждается в постоянном развитии. Развитие данного продукта осуществляется по двум направлениям: наращивание функциональных возможностей (вширь), развитие имеющегося функционала (вглубь).

Последнему направлению стоит уделять особое внимание, так как оно определяет качество программного продукта. На сегодняшний день можно выделить три основных аспекта развития имеющегося функционала: повышение скорости работы при обработке большого количества данных, развитие масштабируемости среды и оптимизация кода, позволяющая максимально расширить круг решаемых с помощью среды задач.

5. Заключение

В данной работе рассмотрены программная реализация и основные функции

среды комплексного анализа производительности алгоритмов балансировки в параллельном методе ветвей и границ.

Среда комплексного анализа производительности параллельных алгоритмов оптимизации является удобным и многофункциональным расширяемым средством для исследования алгоритмов балансировки нагрузки процессоров при решении задач глобальной оптимизации методом ветвей и границ. Часто реализация нового функционала среды требует научного подхода. Данное программное средство позволяет осуществлять как визуальный, так и автоматизированный анализ производительности алгоритмов.

Литература

1. М. А. Посыпкин, И. Х. Сигал, Исследование алгоритмов параллельных вычислений в задачах дискретной оптимизации ранцевого типа // Ж. вычисл.матем. и матем. физ., 2005, том 45, номер 10, С. 1801–1809.
2. И. Х. Сигал, Я. Л. Бабинская, М. А. Посыпкин Параллельная реализация метода ветвей и границ в задаче коммивояжера на базе библиотеки BNB-Solver комплексах // Труды ИСА РАН 2006. Т. 25, С.26-36.
3. Distributed Computing and Its Applications. // Felicity Press, Bristol,USA, 2005.ISBN: 0-931265-10-2, 298p. Монография (соавторы: S.V. Emelyanov, A.P. Afanasiev, Y.R. Grinberg, V.E. Krivtsov, B.V. Peltsverger, O.V. Sukhoroslov, R.G. Taylor, V.V. Voloshinov).
4. Лупин С. А., Посыпкин М. А. Технологии параллельного программирования: Учеб. пос //Сер. Высш. образ-ние. М.: Форум Инфра-М. – 2008. – Т. 208. – С. 2000.
5. Gendron B., Crainic T. G. Parallel branch-and-branch algorithms: Survey and synthesis //Operations research. – 1994. – Т. 42. – №. 6. – С. 1042-1066.
6. Стронгин Р. Г., Гергель В. П., Баркалов К. А. Параллельные методы решения задач глобальной оптимизации //Известия высших учебных заведений. Приборостроение. – 2009. – Т. 52. – №. 10. – С. 25-33.
7. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. БХВ-Петербург. 2002. С. 329.
8. Рейтинг по состоянию на июнь 2015 года с официального сайта Топ 500: <http://www.top500.org/lists/2015/06/>.
9. Страница проекта BNB-Simulator <https://github.com/fominandrey/bnb-simulator>.
10. Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization //Computer Science-Research and Development. 2009. Т. 23. №. 3-4. С. 211-215.
11. Страница проекта BNB-Solver <https://github.com/mposypkin/BNB-solver>.
12. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. БХВ-Петербург. 2002. С. 82-83

ОБЗОР АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ В ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ МЕТОДА ВЕТВЕЙ И ГРАНИЦ⁶

Голубева Я.В.

⁶ Работа выполнена при поддержке РФФИ (проекты № 16-07-00873 А и № 16-07-00458 А).

Аннотация

Статья посвящена проблеме балансировки нагрузки процессоров на примере параллельной реализации метода ветвей и границ. Рассмотрены некоторые стратегии балансировки нагрузки.

1. Введение

Одним из основных методов решения задач дискретной оптимизации является метод ветвей и границ[1]. Идея этого метода заключается в том, что множество допустимых решений разбивается на подмножества, и те подмножества (подзадачи), в которых оптимальных решений не существует, отбрасываются.

Метод ветвей и границ имеет древовидную структуру, где вершины – это полученные в результате разбиений подзадачи (см. рис. 1). Вершина, отмеченная буквой Т – это отбрасываемая вершина, а вершины, отмеченные буквой К – это задачи-кандидаты, которые могут привести к оптимальному решению.

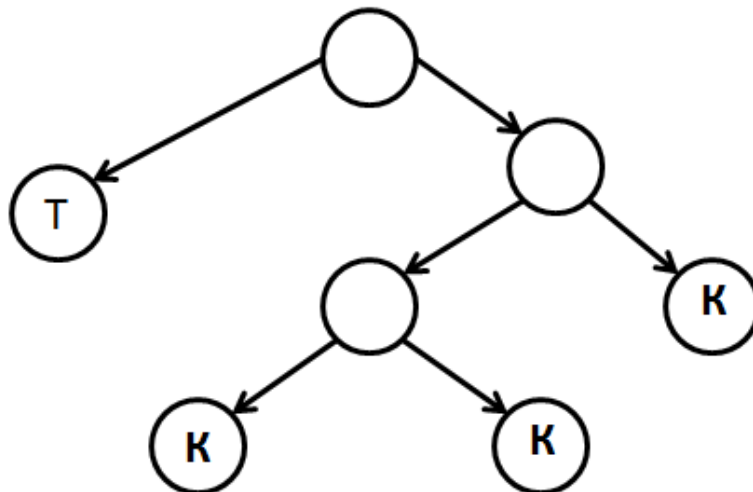


Рис. 1. – Древовидная структура Метода ветвей и границ

Такое представление метода создает благоприятные условия для его параллельной реализации, так как поиск оптимальных решений в отдельных ветвях этого дерева может производиться независимо на разных вычислительных единицах. В то же время при параллельной реализации возникают определенные сложности. В частности, возникает проблема балансировки нагрузки. Она заключается в том, что невозможно статически распределить нагрузку между процессорами, так как дерево является неоднородным и его структура не известна заранее. Это может привести к тому, что в определенный момент времени одни процессоры уже закончат выполнять разветвления и будут простаивать, а другие всё еще будут продолжать решать выделенные им подзадачи.

Таким образом, возникает задача балансировки нагрузки – перераспределения вычислительной нагрузки между параллельно работающими процессорами. Актуальность данной задачи подтверждается тем, что в современных вычислительных устройствах число процессоров растет, а чем больше процессоров – тем сложнее балансировать их нагрузку.

2. Краткий обзор методов балансировки нагрузки

В работах [2,3] рассмотрен достаточно простой алгоритм балансировки нагрузки. Балансировка нагрузки в этом алгоритме централизована: среди всех процессоров выделен управляющий процессор (УП) и рабочие процессоры (РП). На рис. 2 изображена упрощенная схема работы алгоритма.

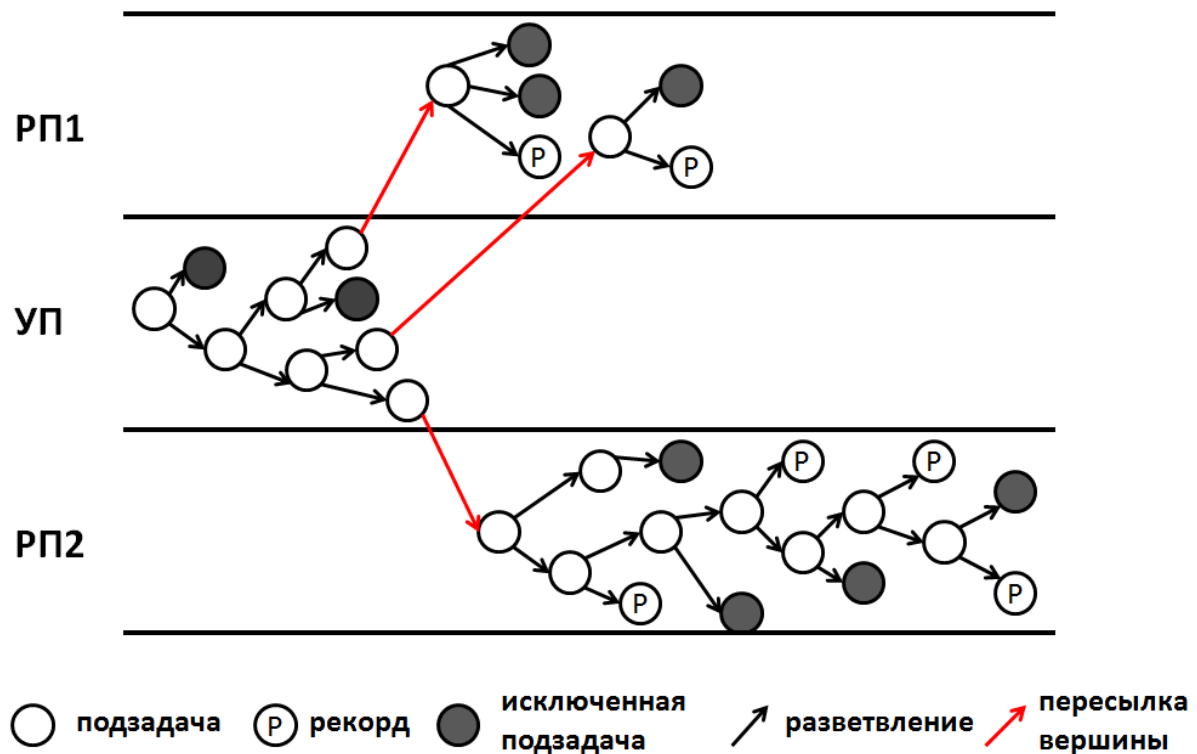


Рис. 2. – Схема стандартного алгоритма параллельной балансировки нагрузки

На начальном этапе УП выполняет заранее определенное количество разветвлений, генерируя некоторое количество подзадач, заведомо большее, чем количество рабочих процессоров. Далее сгенерированные подзадачи распределяются между рабочими процессорами, которые решают их до конца. Если рабочий процессор решил задачу, а на управляющем процессоре еще имеются другие подзадачи – то освободившийся рабочий процессор получает новую подзадачу для решения. Полученные в процессе решений рекорды (наилучшие найденные решения) РП отправляет УП, который, сопоставляя полученные результаты, выбирает оптимальный.

В работе [4] рассматриваются 2 параллельных метода обхода дерева, которые можно использовать в алгоритме централизованной балансировки нагрузки:

1. *Метод назначаемых поддеревьев* – заключается в разбиении дерева на большое число «маленьких» поддеревьев и назначении их для обхода в порядке освобождения процессоров. На первом этапе управляющий процессор осуществляет обход дерева в ширину до тех пор, пока число вершин в очереди

не достигнет заранее заданного числа, заведомо намного большего, чем число процессоров в системе. На втором этапе управляющий процессор рассылает рабочим процессорам вершины из очереди, и они осуществляют их обход в глубину;

2. *Метод выделяемых поддеревьев* – отличается от предыдущего тем, что разбиение на подзадачи происходит не сразу, а по мере выполнения обхода дерева. На первом этапе управляющий процессор осуществляет обход дерева в ширину до тех пор, пока число вершин в очереди не достигнет заранее заданного числа, равного или немного большего, чем число процессоров в системе. На втором этапе управляющий процессор рассылает рабочим процессорам вершины из очереди, и они осуществляют их обход в глубину. По окончании обхода процессор освобождается. Для его загрузки выбирается поддерево, обход которого в этот момент обрабатывается одним из процессоров, и в нем выделяется поддерево, которое передается для обхода освободившемуся процессору.

В работе [5] рассматриваются статическая и динамическая стратегии балансировки нагрузки. В обеих стратегиях сначала генерируется некоторое число подзадач. *Статическая стратегия* подразумевает хранение всех сгенерированных и нерешенных подзадач в общем рабочем пуле. Каждый процессор последовательно решает свою подзадачу до конца, используя свой локальный рабочий пул. *Динамическая стратегия* подразумевает обмен подзадачами между локальными рабочими пулами во время выполнения программы.

Авторы выделяют 3 класса динамических стратегий:

1. *Стратегия балансировки по запросу (Strategy on request)* – процессор с пустым (или с почти пустым) рабочим пулом отправляет запрос другому процессору на получение новых подзадач. Запрос может быть одобрен – в этом случае подзадачи пересылаются из одного рабочего пула в другой – или отклонен;
2. *Балансировка без запроса (Strategy without request)* – процессоры рассылают часть своих подзадач другим процессорам без запроса с их стороны;
3. *Комбинированная стратегия (Combined strategy)* – комбинация двух предыдущих стратегий.

В работе [6] описана общая модель динамической балансировки нагрузки, которая состоит из четырех этапов:

1. *Оценка загрузки процессора (Processor Load Evaluation)* – на этом этапе производится оценка нагрузки на каждом процессоре в системе и выявление разбалансированности;
2. *Определение выгоды балансировки нагрузки (Load Balancing Profitability Determination)* – на этом этапе определяются накладные расходы, которые понесет балансировка нагрузки, и принимается решение, будет ли выгодна балансировка нагрузки в данный момент времени;
3. *Стратегия перемещения подзадач (Task Migration Strategy)* – здесь определяются источники и получатели подзадач для решения;
4. *Стратегия выбора подзадач (Task Selection Strategy)* – на заключительном этапе происходит отбор и отправка подзадач от процессора с высокой нагрузкой процессору с низкой нагрузкой.

В данной модели авторы выделяют следующие 5 стратегий балансировки нагрузки.

1. *Градиентная модель (The Gradient Model)*. Стратегия заключается в том, что процессоры с низкой нагрузкой оповещают другие процессоры о своем статусе, и перегруженные процессоры отвечают отправкой некоторого числа подзадач соседним, менее нагруженным процессорам в системе. Подзадачи перемещаются до тех пор, пока нагрузка соседних процессоров меньше нагрузки процессора-отправителя. Для выполнения распределения излишков нагрузки используется «карта приближений» (см. рис. 3), в которой каждому процессору присваивается число, означающее минимальное количество промежуточных процессоров до менее нагруженного процессора. Если нагрузка на процессоре совсем мала – то ему присваивается 0;

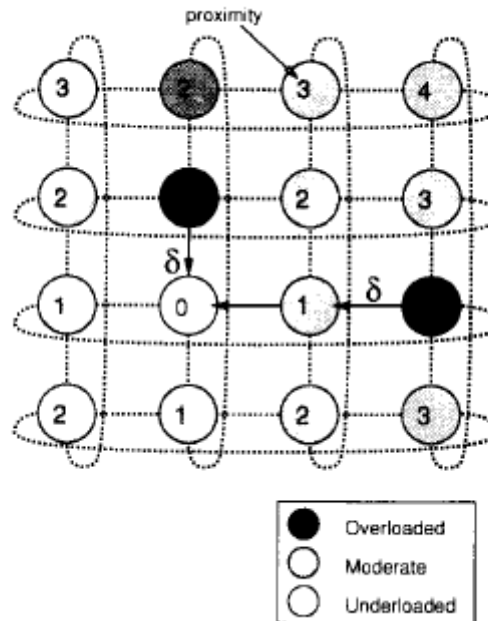


Рис. 3. – «Карта приближений»

2. *Метод распределения, инициированного отправителем (Sender Initiated Diffusion)*. Стратегия заключается в том, что процессор, при получении информации о том, что среди соседних процессоров присутствуют такие, нагрузка которых опустилась ниже некоторого порога, инициирует отправку излишков малонагруженным процессорам;
3. *Метод распределения, инициированного получателем (Receiver Initiated Diffusion)*. Стратегия отличается от предыдущей тем, что отправку излишков инициируют малонагруженные процессоры;
4. *Метод иерархической балансировки (Hierarchical Balancing Method)*. В этой стратегии процессоры выстраиваются в дерево, которое подвергается балансировке нагрузки. Балансировка вызывается на разных уровнях получением сообщения о разбалансированности между процессорами, находящимися одним уровнем ниже. Если степень разбалансированности превышает некоторый порог, то на этом уровне происходит передача излишка подзадач от более нагруженного процессора к менее нагруженному;
5. *Метод измерений (The Dimension Exchange Method)*. Стратегия заключается в то, что N процессоров представляется в виде $\log N$ измерений. Сначала балансируются между собой пары процессоров, находящиеся в первом измерении (это такие процессоры, у которых адреса отличаются только последним значащим битом). Далее балансируются между собой пары процессоров, находящиеся во

втором измерении и так далее, пока каждый процессор не сбалансирует свою нагрузку со всеми своими соседями. На рис. 4 представлена схема данной стратегии для восьми процессоров.

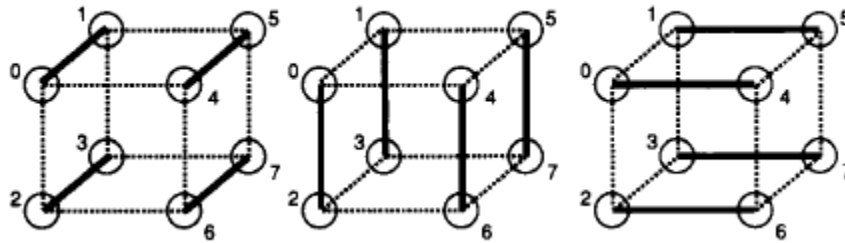


Рис. 4. – The Dimension Exchange Method

В работе [7] представлен алгоритм балансировки нагрузки, который стремится к тому, чтобы максимальная разница между числом подзадач на процессоре и числом подзадач на соседних с ним процессорах меньше некоторого числа Δ :

$$\max_{i \in \{1, \dots, k\}} \frac{|w(p) - w(p_i)|}{|w(p)|} < \Delta,$$

где $w(p)$ – количество подзадач на процессоре, $w(p_i)$ – количество подзадач на i -том соседнем процессоре из k .

Очевидно, что чем меньше параметр Δ – тем лучше становится балансировка нагрузки процессоров. Однако слишком малые значения параметра Δ приводят к тому, что интенсивные коммуникации, связанные с балансировкой нагрузки, останавливают решения подзадач.

Особенностью данного алгоритма является то, что в нем выделены 2 процесса: процесс действия (Actor Process) и процесс контроля (Control Process).

Процесс действия отвечает за балансировку количества подзадач на процессорах и перемещении подзадач с высоконагруженных процессоров на низконагруженные. Процесс контроля отвечает за регулирование параметра Δ . Он увеличивает этот параметр, чтобы сократить время, затрачиваемое на балансировку нагрузки процессоров и дать возможность процессорам возобновить решение имеющихся подзадач, а спустя некоторое время снова уменьшает его для достижения оптимальных результатов балансировки нагрузки.

Авторы приводят результаты экспериментальных исследований на вычислительной системе из 256 процессоров, которые демонстрируют достаточно высокую эффективность данного алгоритма: от 0,872 до 0,973.

3. Заключение

В статье затронута проблема динамического распределения нагрузки между процессорами, возникающая в параллельной реализации метода ветвей и границ. Рассмотрены некоторые алгоритмы балансировки, предложенные для ее решения.

Литература

1. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование. М.: Физматлит, 2002.
2. М.А.Посыпкин, Архитектура и программная организация библиотеки для решения задач дискретной оптимизации методом ветвей и границ на многопроцессорных вычислительных комплексах// Труды ИСА РАН, Т. 25, с. 18-25.

3. Посыпкин М.А., Сигал И.Х. Исследование алгоритмов параллельных вычислений в задачах дискретной оптимизации ранцевого типа // Ж. вычисл. матем. и матем. физ. 2005. Т. 45. № 10. С. 1801-1809.
4. Н. Е. Тимошевская, Параллельные методы обхода дерева, Матем. моделирование, 2004, том 16, номер 1, 105-114.
5. Gendron B., Crainic T.G., Parallel Branch-and-Bound Algorithms: Survey and Synthesis// Operations Research, Vol. 42, No. 6 (Nov. - Dec., 1994), 1042-1066.
6. Willebeek-LeMair, M.H., Reeves, A.P., Strategies for dynamic load balancing on highly parallel computers// Parallel and Distributed Systems, IEEE Transactions on, Vol. 4, No. 9, September 1993, 979-993.
7. Lüling R., Monien B., Load balancing for distributed branch & bound algorithms// Parallel Processing Symposium, 1992. Proceedings., Sixth International, 543-548.

ПРИМЕНЕНИЕ АППАРАТНОГО УСКОРЕНИЯ НА БАЗЕ ТЕХНОЛОГИИ CUDA В ПРОЕКТАХ ДОБРОВОЛЬНЫХ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

Шутов И.В.

Национальный исследовательский технологический университет «МИСиС», кафедра инженерной кибернетики

119049, г.Москва, Ленинский проспект, д.4, vovantuzz@gmail.com

Курочкин И.И.

Институт проблем передачи информации РАН,
127051, г. Москва, Большой Каретный переулок, д.19 стр. 1,
kurochkin@iitp.ru

Аннотация

Добровольные распределенные вычисления являются одним из наиболее популярных способов решения сложных вычислительных задач. Производительность проектов добровольных распределенных вычислений напрямую зависит от популярности проекта[1], спектра поддерживаемых вычислительных устройств, в том числе специализированных вычислителей, а так же балансировки нагрузки на вычислительные узлы. Приведены основные этапы разработки вычислительного приложения для инфраструктуры BOINCс поддержкой вычислений на GPU.

1. Введение

Современные научные задачи для своего решения зачастую требуют большого количества вычислений. Одним из наиболее популярных подходов для решения таких задач является распределение обработки между большим числом вычислителей. В то же время не все научные организации могут позволить себе создание или аренду высокопроизводительных суперкомпьютеров. В случае, если задача может быть разбита на несколько независимых подзадач, то их обработка может быть произведена на грид-системе, в том числе построенной из персональных компьютеров[2]. Добровольными распределенными вычислениями называется способ обработки вычислительных задач путем распределения их на множестве вычислителей грид-системы построенной из персональных компьютеров добровольцев.

Наиболее важным способом повышения производительности проекта добровольных вычислений является расширение спектра поддерживаемого оборудования, которое способно обрабатывать подзадачи. В частности, на сегодняшний день одним из наиболее перспективных направлений является вычисления общего назначения на графических ускорителях или GPGPU. Это является особенно актуальным в рамках проектов добровольных распределенных вычислений за счет широкой распространенности графических ускорителей с необходимыми аппаратными возможностями среди персональных компьютеров добровольцев.

На сегодняшний день существует целый спектр различных инструментов разработки приложений с поддержкой вычислений на GPU. В рамках данной работы было решено использовать API CUDA, разработанное компанией NVIDIA. В пользу данного выбора было учтено несколько факторов, и в частности ситуация на рынке графических ускорителей. По данным аналитической компании MercuryResearch по состоянию на второй квартал 2015 года, графические решения NVIDIA занимали более 80 процентов рынка. Более того, компания обладает рядом профессиональных решений Tesla, которые специализируются на неграфических вычислениях.

2. Предварительное тестирование

В настоящее время вычисления на GPU и, в частности, с использованием технологии CUDA нашли широкое применение в добровольных распределенных вычислениях, где они показывают значительный прирост производительности. Примерами успешного применения GPU при расчетах являются проекты добровольных распределенных вычислений SETI@home[3], GPUGrid[4], Einstein@home[5], MilkyWay@home[6]. Так же сравнительное тестирование GPU и CPU проводилось в ряде исследовательских работ[7].

Тем не менее, на начальном этапе выполнения работы было решено провести тестирование скорости работы CUDA приложений в рамках инфраструктуры BOINC на базе простейших тестовых программ. Было разработано два тестовых приложения:

1) Приложение тестирования скорости сортировки целых чисел, которое базировалось на алгоритме RadixSort (CPU версия использовала QuickSort). Оно показало прирост скорости при обработке на GPU приблизительно в 10 раз. Данные алгоритмы имеют разную сложность сортировки, однако выбор был продиктован целью использовать минимальное количество сторонних библиотек. Таким образом, были использованы стандартные реализации алгоритмов сортировки `std::sort()` и аналог `std` для GPU поставляемый вместе с CUDASDK - `thrust::sort()`.

2) Приложение для тестирования скорости генерации псевдослучайных чисел при помощи вихря Мерсенна. Для GPU версии была использована библиотека CURAND поставляемая вместе с CUDASDK, для CPU использовалась библиотека Boost. Оно показало прирост скорости при обработке на GPU в среднем в 12 раз.

3. Реализация приложения поиска целочисленных параллелепипедов

В роли базовой задачи для исследования возможностей применения аппаратного ускорения в проектах добровольных распределенных вычислений была выбрана задача поиска множества параллелепипедов с целочисленными сторонами, внутренними диагоналями и диагоналями граней. Данная задача может решаться путем перебора раз-

личных комбинаций параллелограммов, образующих грани параллелепипеда, это обуславливает хорошую переносимость алгоритма на GPU[8].

В качестве входных параметров данного приложения являются три множества предварительно сгенерированных параллелограммов, отвечающих условиям целочисленности сторон и диагоналей. Приложение производит полный перебор входных массивов данных и проверяет всевозможные комбинации на соответствие заданным условиям целочисленности.

Версия приложения, предназначенная для обработки на центральном процессоре, предполагает полный последовательный перебор входных параллелограммов.

GPU-версия приложения реализует массивно-параллельную обработку входных данных путем запуска одномерной сети потоков. При каждом запуске GPU-ядра фиксируется одна из граней параллелепипеда, номер каждого запущенного потока представляет собой индекс параллелограмма второй грани параллелепипеда. Каждый поток ведет перебор всех возможных параллелограммов третьей грани.

Приложения были реализованы при помощи языка программирования C++ с использованием CUDA SDK 7.0.

Реализованные приложения были внедрены в тестовый BOINC-проект, развернутый на базе облачной инфраструктуры. Тестирование на различных конфигурациях оборудования показало существенный прирост скорости обработки заданий при использовании возможностей аппаратного ускорения.

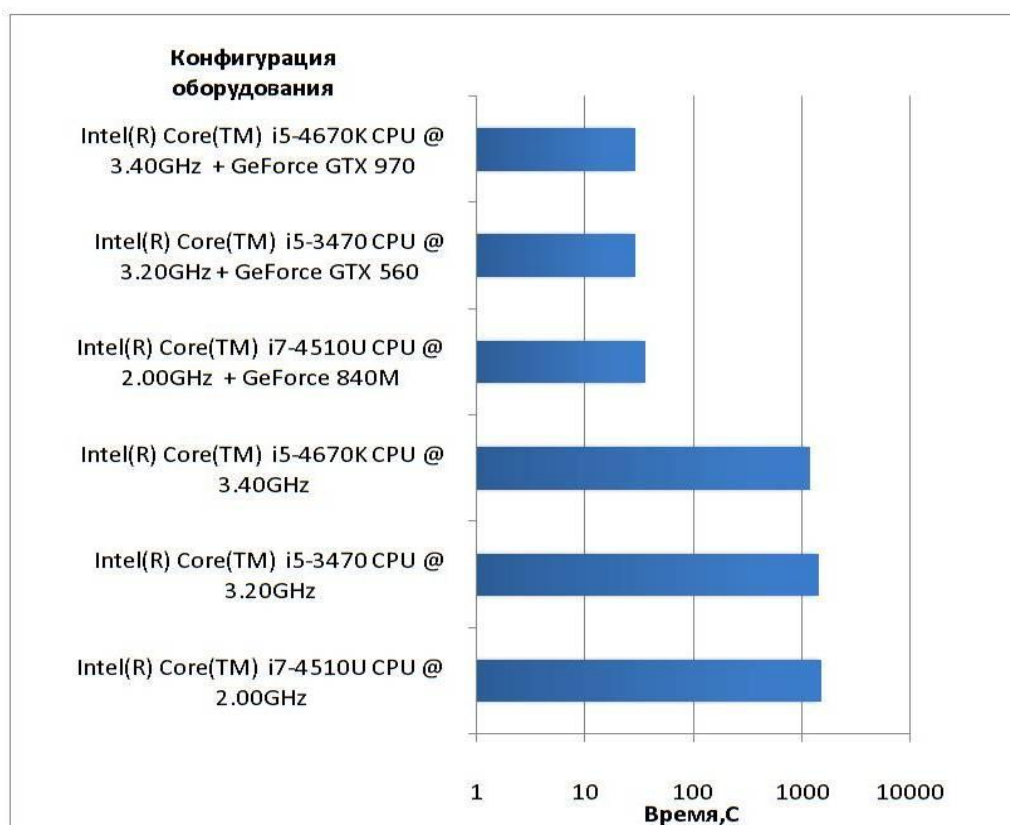


Рис. 1. — Сравнительное тестирование приложений для CPU и GPU

4. Заключение

Была проанализирована возможность применения аппаратного ускорения в проектах добровольных распределенных вычислений. Было реализовано несколько тестовых приложений, которые показали целесообразность применения графических ускорителей в проектах добровольных вычислений. После этого, в интересах международного проекта добровольных распределенных вычислений Optima@home были реализованы CPU и GPU версии переборного приложения поиска параллелепипедов с целочисленными сторонами и диагоналями. Данные приложения были внедрены в тестовый проект, где приложение с поддержкой аппаратного ускорения в среднем показало 44-кратное увеличение скорости обработки.

В дальнейшем планируется исследование методик балансировки нагрузки на вычислительные узлы с GPU, с целью сокращения общего времени обработки задачи и увеличения надежности вычислений.

Работа выполнена при поддержке РФФИ (грант 16-07-00659 А).

Литература

1. Якимец В.Н., Курочкин И.И. Добровольные распределенные вычисления в России: социологический анализ // Труды XVIII объединенной конференции «Интернет и современное общество» (IMS-2015). Университет ИТМО; Библиотека Российской академии наук. Санкт-Петербург, 2015.
2. О.С. Заикин, М.А. Посыпкин, А.А. Семёнов, Н.П. Храпов. Организация добровольных вычислений на платформе BOINC на примере проектов OPTIMA@home и SAT@home // CAD/CAM/CAE Observer # 3 (71) / 2012.
3. Официальный сайт проекта добровольных распределенных вычислений SETI@home. URL://<http://setiathome.ssl.berkeley.edu/>
4. Официальный сайт группы проектов GPUGrid. URL: <https://www.gpugrid.net/>
5. Официальный сайт международного проекта Einstein@home. URL: <https://einstein.phys.uwm.edu/>
6. Официальный сайт международного проекта MilkyWay@home. URL: <http://milkyway.cs.rpi.edu/milkyway/>
7. Lee, V. W. et al. Debunking the 100x gpu vs. cpu myth: an evaluation of throughput computing on cpu and gpu. In Proc. ISCA, 2010
8. Jorge F. Sawyer, Clifford A. Reiter, Perfect Parallelepipeds Exist. Lafayette College, Easton

АНАЛИЗ СЕРВИС-ОРИЕНТИРОВАННЫХ СИСТЕМ: ОБНАРУЖЕНИЕ АНТИПАТТЕРНОВ

Югов Александр Сергеевич

Национальный исследовательский университет «Высшая Школа Экономики»
Москва, yugovas@live.ru

Аннотация

Сервис-ориентированные системы, как и любые другие программные системы,

развиваются с течением времени. Независимо от того, какими были предпосылки изменений: новые требования, изменение среды функционирования, и т.п. Постоянные изменения могут привести к появлению в системе «плохих» решений – антипаттернов, что, в свою очередь, снижает качество программной системы и требует большего внимания разработчиков при поддержке и дальнейшем развитии продукта. В данной статье рассмотрены примеры антипаттернов, а также способы их автоматизированного нахождения с помощью построения модели реального поведения системы

7. Введение

Сервис-ориентированные системы, как стиль в архитектуре приложений, очень широко принят в промышленной разработке программного обеспечения, потому что это позволяет разрабатывать гибкие и масштабируемые распределенные системы по более выгодной цене. Результатом разработки становятся автономные, многоразовые и независимые от платформы использования функционала единицы – сервисы – которые могут потребляться через Интернет [9].

Традиционные подходы к разработке основаны на фазах жизненного цикла развития системы, когда в процессе разработки принимают участие различные команды или даже различные компании [10]. Кроме того, в классических подходах, акцент делается на одного поставщика, разрабатывающего целиком систему или подсистему. Появление сервис-ориентированного подхода вводит модель архитектуры, разделенной на уровни. Это дает возможность существования различных подходов разработки, в результате чего различные участники поставляют отдельные элементы слоев сервис-ориентированных систем. Опыт разработки таких совместных проектов, разделенных на отдельные сервисы, показывает, что могут появляться потенциально ошибкоопасные участки. В рамках данной работы будем называть их анти-паттерны.

Анти-паттерны в процессе эксплуатации систем на базе сервисов представляют собой «плохие» решения повторяющихся проблем проектирования. В противоположность паттернам проектирования, которые являются хорошими проверенными решениями, анти-паттерны инженерам следует избегать. Анти-паттерны также могут быть введены как следствие различных изменений, таких как, например, новые требования пользователей или изменения среды функционирования.

Далее в работе представлены введение в сферы обнаружения антипаттернов. После чего описан предлагаемый подход для автоматизированного обнаружения антипаттернов.

8. Примеры антипаттернов в сервис-ориентированных системах

Качество дизайна (архитектуры) имеет жизненно важное значение для построения продуманной, удобной в обслуживании и эволюционирующей системы. Наличие в системе паттернов проектирования, как и антипаттернов, было признано одним из наиболее эффективных способов выражения архитектурных проблем и их решений, и, следовательно, критерием более высокого качества различных систем [23].

Кроме того, был предпринят ряд усилий по формализации свойств понятия «плохой» практики, т.е. решения, которые отрицательно влияют на качество системы. Некоторые совершенно новые подходы были введены для определения и обнаружения уязвимостей кода и анти-паттернов [22], [14]. Методы, используемые в этих подходах, могут быть очень разнообразными: полностью ручные, на основе руководств исследования; метрические, основанные на эвристических методах, с использованием правил и пороговых уровней для различных метрик; или на основе Байесовских сетей.

Некоторые подходы [1] применимы на уровне приложений и могут быть применены на начальных стадиях жизненного цикла программного обеспечения.

Довольно большое количество методологий и инструментов существуют для обнаружения антипаттернов, в частности, в объектно-ориентированных (ОО) системах [13], [16]. Однако, обнаружение анти-паттернов в сервис-ориентированных системах, в отличие от ОО систем, все еще находится на стадии зарождения. Одна из последних работ по обнаружению антипаттернов в сервис-ориентированных архитектурах (СОА) была предложена Moha et. al. в 2012 [22].

Авторы предложили подход к определению и обнаружению обширного набора антипаттернов в СОА, которые охватывают такие понятия, как зернистость (*granularity*), связность (*cohesion*) и дублирование (*duplication*). Их инструмент способен обнаружить наиболее популярные СОА анти-паттерны, определенные в литературе. Помимо этих антипаттернов, авторами определены еще три антипаттерна, а именно: узкое место системы (*bottleneck service*), цепь сервисов (*service chain*) и сервис данных (*data service*). Узкое место – это сервис, который используется многими другими компонентами системы, и, как следствие, характеризуется высокой входящей и исходящей связностью, влияющей на время отклика сервиса. Цепи сервисов появляется, когда бизнес-задача решается с помощью длинной цепочки последовательных вызовов. Сервис данных является службой, которая выполняет простые операции поиска информации или доступа к данным, которые могут повлиять на связность компонент.

В 2012 году Rotem-Gal-Oz [26] определил антипаттерн «узел» (*knot*), как набор мало связанных сервисов, которые, тем не менее, тесно зависят друг от друга. Этот антипаттерн может привести к снижению удобства использования и увеличению времени отклика системы.

Другой пример антипаттерна, «песочница» (*sand pile*) определенный Kr'Al и др. [15] появляется, когда многие мелкие сервисы используют общие данные, которые могут быть доступны через сервис, представляющий собой антипаттерн «сервис данных».

В работе Cherbakov и др. предложен антипаттерн «дубликат» [5] который влияет на совместное использование сервисы, содержащие схожие функции, что может вызвать проблемы в процессе поддержки.

В 2003 Dudney и др. [8] определили набор анти-паттернов для J2EE-приложений. Среди прочих выделяются антипаттерны «множественный сервис» (*multi service*), крошечный сервис (*tiny service*) и «болтливая парочка» (*chatty service*). Множественный сервис – это сервис, который предоставляет различные бизнес-операции, которые не обладают практической схожестью (например, относятся к различным подсистемам), что может повлиять на доступность сервиса и время отклика. Крошечный сервис – небольшая служба с несколькими методами, которые всегда используются вместе. Этот антипаттерн может повлечь невозможность повторного использования. И, наконец, анти-паттерн «болтливая парочка» представляет сервисы, которые постоянно вызывают друг друга, передавая небольшие объемы информации.

9. Подход к обнаружению антипаттернов, основанный на метриках

Как отмечал De Marco, [7] для того, чтобы контролировать качество разработки, необходимо правильные количественные методы. Уже в 1990 году Card подчеркивал, что метрики должны быть использованы для оценки разработки программного обеспечения с точки зрения качества [3]. Но что стоит измерять? В вышеупомянутом контексте проектных норм, принципов и эвристик вопрос должен быть перефразирован

следующим образом: можно ли выразить принципы «хорошего дизайна» в измеримой форме?

Основная цель этого подхода заключается в том, чтобы предоставить инженерам механизм, который позволит им работать с метриками на более абстрактном уровне, что концептуально гораздо ближе к реальным условиям применения численных характеристик. Механизм, определенный для этой цели, называется стратегией обнаружения:

Стратегия обнаружения – это количественное выражение правил, с помощью которых специфичные фрагменты программного продукта (элементы архитектуры), соответствующие этому правилу могут быть обнаружены в исходном коде.

Поэтому стратегия обнаружения является общим механизмом для анализа модели исходного кода с помощью метрик. Следует отметить, что в контексте предыдущего определения, «количественное выражение правила» означает, что правило должно быть надлежащим образом выразимо с помощью метрик. Использование метрик в стратегиях обнаружения обосновано механизмами фильтрации и композиции. В следующих подразделах эти два механизма будут рассмотрены более подробно.

Ключевой проблемой в фильтрации данных является задача уменьшение исходного набора информации таким образом, что сохраняются только те значения, которые представляют собой особую ценность. Это, как правило, называют редукцией данных [12]. Целью является обнаружение таких элементов системы, которые обладают особыми свойствами. Пределы (границы) подмножества определяются на основании типа фильтра данных. В контексте процесса измерения применительно к программному обеспечению, мы обычно пытаемся обнаружить крайние (аномальные) значения или такие значения, которые находятся в определенном диапазоне. Поэтому выделяют следующие типы фильтров [19]:

- **Предельный фильтр** – фильтр данных, в котором один предел (граница) в результирующем наборе явно отождествляется с соответствующим ограничением исходного набора данных.
- **Интервальный фильтр** – фильтр данных, в котором как нижний, так и верхний пределы полученного подмножества явно указаны в определении набора данных.

Предельный фильтр. В зависимости от того, как мы указываем предел(ы) результирующего набора данных предельные фильтры могут быть семантическими или статистическими.

1. **Семантические.** Для этих фильтров должны быть указаны два параметра: пороговое значение, которое указывает на предельное значение (должно быть явно указано); и направление, что определяет, является ли пороговое значение верхним или нижним предела отфильтрованного набора данных. Эта категория фильтров называется семантической, так как выбор параметров основан на семантике конкретной метрики, в рамках выбранной модели интерпретации для этой метрики.
2. **Статистические.** В отличие от семантических фильтров, статистические, не нуждаются в явной спецификации для порогового значения, так как она определяется непосредственно из исходных данных, установленных с использованием статистических методов (например, диаграммы разброса). Тем не менее, направление по-прежнему должно быть указано. Статистические фильтры строятся исходя из предположения, что все измеренные подразделения системы структурно разработаны с использованием одно и того же стиля и, следовательно, результаты измерений сопоставимы.

В данной работе используется набор конкретных фильтров данных из двух предыдущих категорий. На основе их практического использования и выбранной модели интерпретации эти конкретные фильтры могут быть сгруппированы следующим образом:

- Абсолютные семантические фильтры: *HigherThan* и *LowerThan*. Эти механизмы фильтрации параметризуются числовым значением, представляющим границу. Мы будем использовать такие фильтры данных, чтобы выразить «четкие» правила проектирования или эвристики, – например, «сервис не должен быть связан с более чем 6 другими сервисами». Стоит отметить, что порог задается в качестве параметра фильтра, в то время как два возможных направления захватываются двумя конкретными фильтрами.
- Относительные семантические фильтры: *TopValues* и *BottomValues*. Эти фильтры разграничивают отфильтрованные данные, установленные параметром, определяющим количество объектов, которые будут извлечены, а не указывают значение максимального (или минимального) дозволённого значения в наборе результатов. Таким образом, значения в наборе результатов будут рассматриваться относительно исходного набора данных. Используемые параметры могут быть абсолютными (например, «выбрать 20 объектов с наиболее высокими значениями») или процентными (например, «извлечь 10% всех измеренных объектов, имеющих самые низкие значения»). Этот вид фильтров полезен в ситуациях, когда необходимо рассмотреть самые высокие (или низкие) значения из заданного набора данных, а не с указанием точных пороговых значений.
- Статистические: диаграммы разброса. Диаграммы разброса – статистический метод, с помощью которого можно обнаружить аномальные значения в наборе данных [11]. Фильтры данных на основе таких статистических методов, которые, конечно, не ограничиваются диаграммами разброса, полезны в количественной оценке правил. Опять же, мы должны указать направление отклонения прилежащих значений на основе семантики правила проектирования.
- Интервальные Фильтры. Очевидно, для интервала данных необходимо определить два пороговых значения. Однако, в контексте стратегий обнаружения, где в дополнение к механизму фильтрации существует механизм композиции, интервальные фильтры сводятся к композиции двух предельных (абсолютных семантических) фильтров противоположных направлений.

Далее рассмотрим набор правил, которыми должны руководствоваться инженеры при решении, какой тип фильтра данных должен быть применен для фильтрации результатов подсчета конкретной метрики.

1. Абсолютный семантический фильтр стоит использовать при количественной оценке ограничений, которые задаются конкретными пороговыми значениями в явном виде.
2. Относительный семантический фильтр подходит в ситуации, когда ограничения определяются в терминах нечетких предельных значений, таких как «высокие» / «низкие» значения.
3. Для больших систем, как правило, удобнее использовать параметризуемые относительные семантические фильтры (где параметр – процент от исходного набора). И наоборот, абсолютные величины чаще

используются при анализе относительно малых систем.

4. Статистический фильтр рекомендуется использовать для тех случаев, когда ограничения вводятся как экстремально высокие или низкие значения, без указания каких-либо точных границ.

В отличие от простых метрик и их моделей интерпретации, стратегия обнаружения должна быть в состоянии делать выводы на основе целого ряда правил. Как следствие, в дополнение к механизму фильтрации, который поддерживает интерпретацию отдельных метрических результатов, нам нужен второй механизм для сопоставления целого ряда результатов расчётов метрик – это механизм композиции. Механизм композиции представляет собой правила объединения результатов вычисления значений метрик. В литературе выделяются 3 оператора композиции: «and», «or» и «butnot» [19]

Эти операторы можно рассматривать с двух различных точек зрения:

- С логической точки зрения. Эти три оператора являются отражением «точек соприкосновения» правил объединения нескольких стратегий обнаружения, в которых операндами являются описания конструктивных характеристик (симптомов). Они облегчают чтение и понимание стратегии обнаружения, потому что благодаря операторам композиции правило, выраженное в виде количественных характеристик, становится похоже на изначальную неформальную формулировку мысли. С этой точки зрения, например, оператор «and» предполагает наличие у исследуемого объекта как симптома, описанного с левой стороны от него, так и наличие симптома, представленного с правой стороны.
- С точки зрения множеств. Данная точка зрения помогает понять, как построен конечный результат стратегии обнаружения. Исходный набор результатов расчёта каждой из метрик ($M_1 \dots M_n$) проводится через механизм фильтрации. После чего остаётся ограниченный набор элементов системы (и рассчитанных метрик для этих элементов), которые представляют интерес для дальнейшего ($F_1 \dots F_n$). После этого полученное множество отфильтрованных наборов должны быть объединены с помощью операторов композиции. Таким образом, с точки зрения операций над множествами, оператор «and» будет соответствовать операции пересечения (\cap), оператор «или» – операции объединения, оператор «butnot» – операции вычитания.

Сложности применения пороговых значений.

Стоит обратить особое внимание на один из аспектов, который имеет решающее влияние на точность стратегии обнаружения: пороговые значения, используемые в любой стратегии обнаружения. Проблема состоит в том, каковы точные пороговые значения и как их можно найти? Существует три способа ответить на эти вопросы:

1. Опыт предшественников и литература. В большинстве случаев установка порогового значения является весьма эмпирическим процессом, основой которого является аналогичный опыт предыдущих экспериментов, а также советы авторов метрик [17]. В случае, если для конкретной метрики нет специальных указаний к подбору пороговых значений, то рекомендуется устанавливать достаточно слабые ограничения, поскольку предпочтительнее получить несколько больше ложных положительных результатов, чем потерять реальные факты из-за строгости ограничений.
2. Подбор. Перспективный подход можно найти в [21], где автор определяет «машину настройки», которая пытается автоматически подобрать соответствующие пороговые значения, и, таким образом настроить стратегии об-

наружения. Этот подход основан на использовании обучающей выборки, подобранной специалистами. На основании этой выборки «машина настройки» выбирает те пороговые значения, которые максимально точно описывают правила в обучающей выборке. Недостаток этого подхода заключается в том, что хранилище примеров должно быть достаточно большим, чтобы обеспечить надлежащую настройку системы.

3. Анализ нескольких вариантов. Еще один важный подход к этой проблеме заключается в расширении процесса обнаружения путем объединения стратегий, применяемых на одной версии системы с сохранением дополнительной информации об ее истории [24]. Анализ версии может дать, например, информацию об устойчивости компонента к изменениям или сохранении ошибки проектирования с течением времени. Такой подход не предназначен, для подбора пороговых значений, но он повышает точность процесса обнаружения, добавляя каждому «подозрительному» объекту системы временную составляющую, которая действует как дополнительный фильтр.

10. Определение стратегии обнаружения

В этом разделе будет писано формирование стратегии обнаружения на примере конкретного анти-паттерна «Божественный объект» [25]. Отправной точкой считается наличие одного (или более) неформального правила, которое описывает проблемную ситуацию. В данном примере будем исходить из трех эвристик, найденных в книге Riel [25]:

1. Сервисы верхнего уровня должны равномерно разделять ответственность.
2. Сервисы не должны содержать большого количества несвязных функций.
3. Сервисы не должны иметь доступ к данным других сервисов.

Пошаговая стратегия

Первым шагом к построению стратегии обнаружения является преобразование неформальных правил в набор коррелирующих симптомов (например, чрезмерная сложность методов, высокая связность компонентов), которые могут быть отражены одной метрикой. В нашем случае первое правило относится к равномерному распределению ответственности между сервисами. Таким образом, симптомом, показывающим нежелательный эффект, является высокая степень сложности сервиса (симптом отражает неравномерное распределение). Второе правило говорит об уровне внутриклассовой связи между функциями. Таким образом, это относится к низкой связности составляющих (функций) сервиса. Третье эвристическое правило говорит об отсутствии прямого доступа к данным (полям, переменным), определенным в других классах. В этом случае появляется симптом доступности "чужих" данных.

Второй шаг заключается в выборе метрик, которые количественно отражают каждый из определенных свойств. Для анти-паттерна «Божественный объект» эти свойства включают сложность, Связность компонентов и доступа к данным другого сервиса. Поэтому был выбран следующий набор метрик:

- Взвешенная сумма сложности методов. Представляет собой сумму статической сложности всех функций сервиса [6]. В качестве меры сложности была рассмотрена цикломатическая сложность McCabe [20].
- Связность компонентов. Является нормированным числом непосредственно связанных функций [2].
- Доступность «чужих» данных. Представляет собой число внешних

сервисов, у которых данный сервис может получить данные напрямую или через специальные методы [18].

Следующий шаг заключается в нахождении адекватного механизма фильтрации, который должен использоваться с каждой из метрик. Этот шаг в основном осуществляется на основе правил, определенных ранее. Таким образом, для первой метрики «взвешенная сумма сложности методов» выберем относительный семантический фильтр *TopValues*. Для метрики «связность компонентов» выберем также относительный семантический фильтр, но на этот раз *BottomValues*. Для третьей метрики выберем абсолютный фильтр, так как мы необходимо отобрать все компоненты, имеющие доступ к «чужим» данным. Таким образом, будем использовать фильтр *HigherThan*.

Одной из наиболее важных задач в определении стратегии обнаружения является установка параметров (пороговых значений) для каждого фильтра. Ранее были описаны различные подходы к этому вопросу. Установим значение 25% как для фильтра *TopValues* метрики «взвешенная сумма сложности методов», так и для фильтра *BottomValues* метрики «связность компонентов», это означает, что каждого из двух метрик оставит в рассмотрении только одну четверть сервисов системы. Что касается порогового значения для фильтра *HigherThan* метрики «доступность чужих данных»: прямого доступа к данным не должно быть разрешено, так что пороговое значение равно 1.

Последним шагом необходимо соотнести перечисленные симптомы, с помощью операторов композиции. Из эвристического контекста, представленного автором [25] следует, что все эти три симптома должны существовать вместе. Поэтому будем использовать оператор объединения (\cap), т.е. чтобы сервис был назван «божественным объектом», он должен обладать всеми тремя заданными симптомами.

Целью данной работы является использование стратегии обнаружения, чтобы выразить правила, которые помогут выявить проблемы проектирования в сервис-ориентированной программной системе, т.е., чтобы найти те фрагменты архитектуры, которые содержат конкретный анти-паттерн. Однако, механизм стратегии обнаружения и методика целиком не ограничиваются обнаружением проблем. Например, целью исследования может быть обратный инжиниринг (reverse engineering) [4], определение паттернов проектирования [18], идентификация компонентов в устаревших системах [27] и т.д.

11. Реализация инструмента поиска анти-паттернов в сервис-ориентированных системах

Практически во всех упомянутых выше работах обнаружение антипаттернов производилось с помощью расчета метрик. В данной работе для обнаружения антипаттернов также будет использован подход на основе метрик.

Далее представлены метрики, используемые в данной работе:

- (1) количество входящих вызовов сервиса (кто-то вызывает данный сервис);
- (2) количество исходящих вызовов сервиса (данный сервис вызывает кого-то);
- (3) время ответа;
- (4) количество взаимных вызовов;
- (5) связность с другими сервисами; и др.

Для вычисления значений каждой из метрик строится специфическая для нее модель. Конкретные значения, полученные в процессе вычисления показывают, входит ли сервис в антипаттерн или нет.

Базовой моделью для расчета всех метрик является графовая модель потока работ

(рис. 1), который проходит каждый объект в процессе обработки. В рамках данной работы предполагается, что каждый объект, попавший в систему инициирует некоторую последовательность операций, которые будут выполнены над данным объектом. Эту последовательность операций будет называть потоком работ. Стоит отметить, что не каждая сервис-ориентированная система построена по такому принципу, но мы будем рассматривать только такие системы.

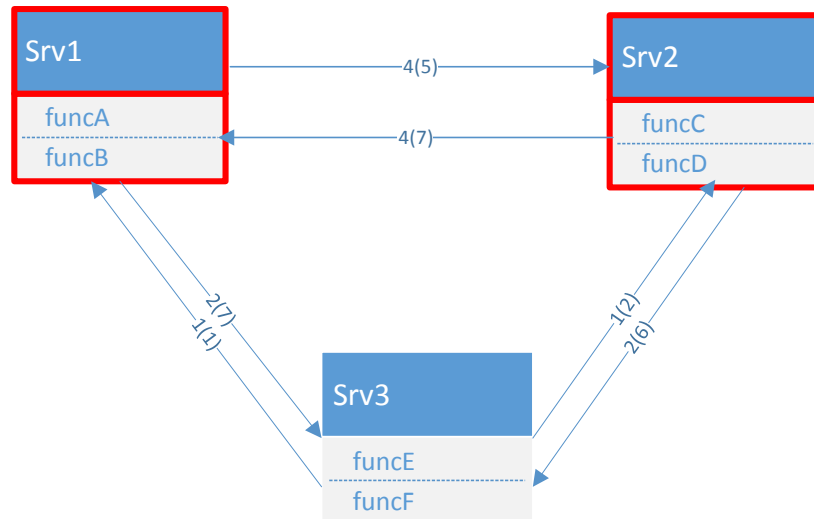


Рис. 1. – Базовая графовая модель для расчета метрик

В данной модели вершинами графа представлены сервисы системы. Дугами графа представлено отношение вызова, т.е. ориентированная дуга от сервиса Srv1 к сервису Srv2 показывает, что сервис Srv1 в процессе функционирования вызывает одну из функций сервиса Srv2. В зависимости от того, какую метрику необходимо рассчитать, ребра графа помечаются специфическими значениями. На примере рис.1 дуги помечены количеством вызовов в конкретном направлении и, в скобках, некоторой взвешенной величиной переданных данных.

Рассмотрим пример антипаттерна Chatty Service. Его подробное описание можно найти по адресу: <http://sofa.uqam.ca/resources/antipatterns.php>. По сути, антипаттерн Chatty Service состоит из нескольких сервисов, которые обмениваются друг с другом большим количеством сообщений малого размера, чаще всего, примитивного типа.

Таким образом, чтобы определить правило для определения антипаттерна Chatty Service, необходимо выделить следующее правило, состоящее из двух метрик и соответствующих им значений:

1. количество взаимных вызовов: высокий уровень;
2. связность с другими сервисами: высокий уровень. Высокий уровень соответствует показателю 80% и выше.

Для автоматизации процесса обнаружения антипаттернов был создан исследовательский прототип информационной системы, реализующей описанный подход. Схема предлагаемого подхода представлена на рис. 2.

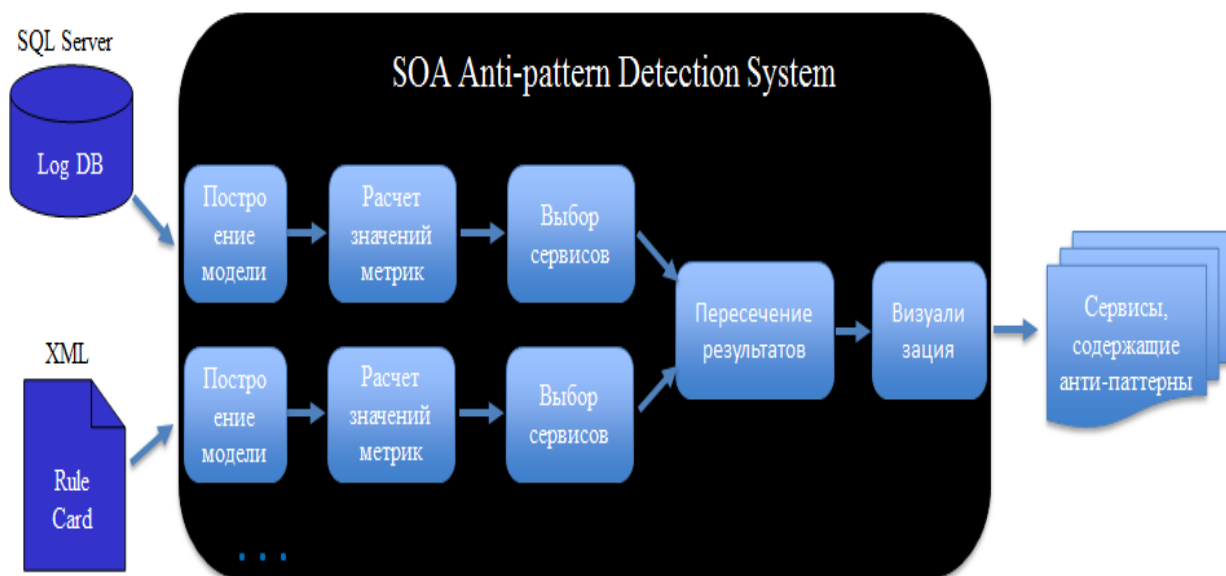


Рис. 2. – Схема подхода для обнаружения анти-паттернов

Рассмотрим правила обнаружения анти-паттернов на основе расчёта метрик. Правила подаются программе на вход в виде XML файлов. Каждое правило содержит ряд метрик с заранее определенными пороговыми значениями. Каждая метрика имеет свой уникальный алгоритм расчета. После выполнения этого алгоритма, анализирующая система получает числовое значение метрики. Это значение сравнивается с правилом, определенным в правиле обнаружения. Если полученное значение совпадает с предопределенным значением, это означает наличие симптома у конкретного сервиса. В противном случае, это означает отсутствие симптома. После чего происходит объединение результатов расчетов согласно заранее выбранной стратегии. На выходе получаем сервисы, которые были выявлены, как подозрительные согласно выбранной стратегии обнаружения.

Для применения стратегии обнаружения на конкретной сервис-ориентированной программной системе, мы должны иметь всю необходимую информацию о самой системе. Эта информация хранится в метамодели, которая состоит из информации об архитектуре (например, классы, методы, переменные) системы, а также о существующих отношениях (например, наследование, вызов методов) между этими объектами (сервисами). Эта модель используется в качестве уровня абстракции от различных языков программирования, технологий реализации и т.п. Точнее, после того, как исходная модель была построена, она хранится в реляционной базе данных.

12. Нахождение антипаттернов

Рассмотрим работу описанного выше инструмента для упоминавшегося ранее антипаттерна “Bottleneck Service”. Журнал событий, подаваемый на вход программы выглядит следующим образом:

Таблица 1. Журнал событий для примера «Bottleneck Service»

Trace Id	Service Name	Function Name	Occurrence Time
1	Srv2	C	03.05.2016 21:22
1	Srv1	A	03.05.2016 21:23

Trace Id	Service Name	Function Name	Occurrence Time
2	Srv4	F	03.05.2016 21:23
3	Srv3	E	03.05.2016 21:24
1	Srv2	C	03.05.2016 21:26
4	Srv4	G	03.05.2016 21:27
3	Srv1	B	03.05.2016 21:27
3	Srv3	E	03.05.2016 21:28
4	Srv1	A	03.05.2016 21:29
4	Srv2	C	03.05.2016 21:30
2	Srv1	A	03.05.2016 21:31
2	Srv2	C	03.05.2016 21:32

Указанный выше журнал показывает, что сервис *Srv1* является общеиспользуемым, т.е. к нему обращаются все остальные сервисы, и его время ответа достаточно велико, т.е. формирует ожидание у других сервисов, что делаем его «проблемным».

Результат, который мы получаем при программной проверке (рис. 3) говорит нам ровно то же самое: сервис *Srv1* обозначен специальным знаком («!»), это означает наличие у него всех симптомов.

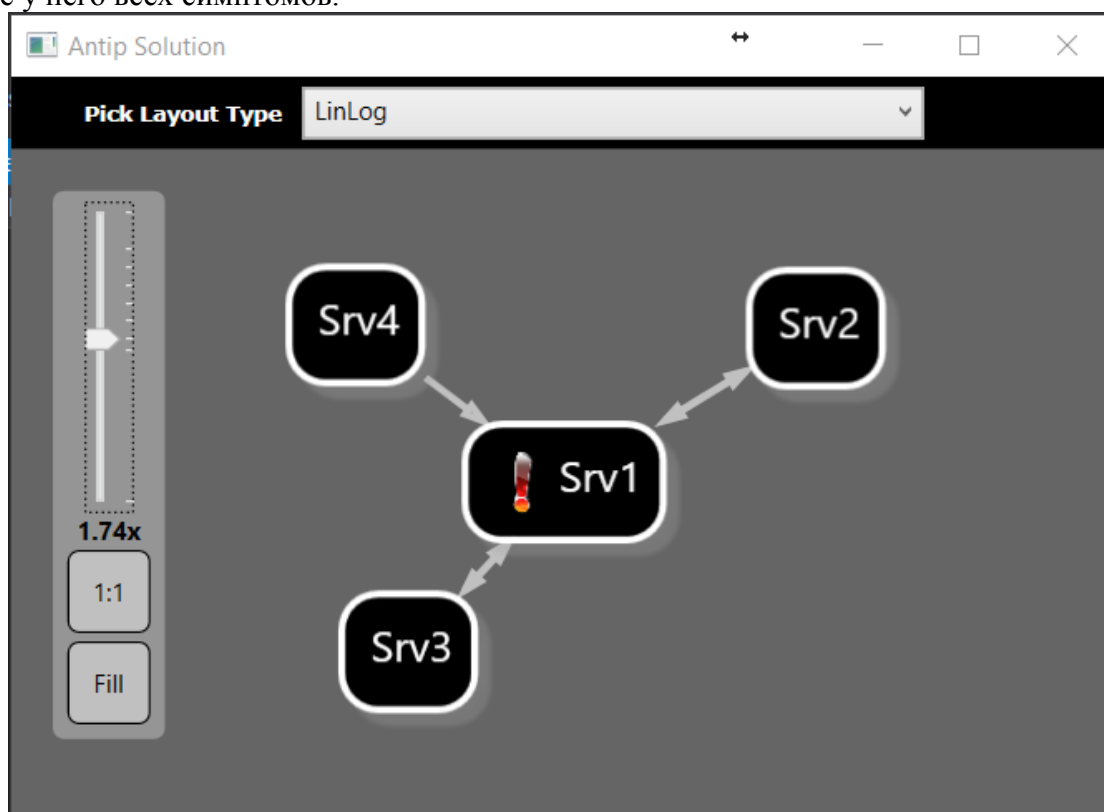


Рис. 3. – Результат обнаружения анти-паттерна «Bottleneck Service».

В дальнейшем планируется апробация инструментария на журналах событий реальных информационных систем

13. Заключение

Данная работа посвящена вопросу необходимости мониторинга состояния СО программных систем в условиях непрерывного развития и совершенствования, когда размер и сложность систем расширяется слишком быстро, чтобы человек мог справиться подручными средствами.

В контексте данной работы был произведен обзор методов для выявления антипаттернов. В ходе исследования были предложены правила обнаружения некоторых антипаттернов. Правила состоят из нескольких метрик и их конкретных значений, описывая симптомы антипаттернов. На текущий момент используется пять метрик: количество входящих и исходящих вызовов, время отклика, связность с другими сервисами. С применением этих метрик были предложены алгоритмы обнаружения нескольких антипаттернов, один из примеров приведен в статье.

Алгоритмы обнаружения антипаттернов были реализованы в виде исследовательского прототипа, который позволяет, задав правила в формате XML, обнаруживать антипаттерны.

В дальнейшем планируется увеличение количества вычисляемых метрик, и нахожимых с их помощью антипаттернов. Кроме того, планируется введение динамического отслеживания изменения значения метрик.

Литература

1. D. Arcelli, V. Cortellessa, C. Trubiani. Experimenting the Influence of Numerical Thresholds on Model-based Detection and Refactoring of Performance Antipatterns. ECEASST 59 (2013).
2. J.M. Bieman and B.K. Kang. Cohesion and Reuse in an Object-Oriented System. Proc. ACM Symposium on Software Reusability, apr 1995.
3. D. Card and R. Glass. Measure Software Design Quality. Prentice-Hall, NJ, 1990.
4. E. Casais. State-of-the-art in Re-engineering Methods. achievement report SOAMET-A1.3.1, FAMOOS, October 1996.
5. L. Cherbakov, M. Ibrahim, J. Ang. Soa antipatterns: the obstacles to the adoption and successful realization of service-oriented architecture.
6. S. R. Chidamber and C. F. Kemerer. A Metric Suite for Object-Oriented Design. IEEE Transactions on Software Engineering, 20(6):476–493, June 1994.
7. T. DeMarco. Controlling Software Projects: Management, Measurement and Estimation. Yourdan Press, New Jersey, 1982.
8. B. Dudney, J. Krozak, K. Wittkopf, S. Asbury, and D. Osborne, J2EE Antipatterns, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.
9. T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, August 2005.
10. G. Farrow. SOA antipatterns: When the SOA paradigm breaks // IBM Developer Works [Online]. Available: http://www.ibm.com/developerworks/library/wa-soa_antipattern/
11. N. Fenton and S.L. Pfleeger. Software Metrics: A Rigorous and Practical Approach. International Thomson Computer Press, London, UK, second edition, 1997.
12. P.G. Hoel. Introduction to Mathematical Statistics. Wiley, 1954.
13. M. Kessentini, S. Vaucher, H. Sahraoui. Deviance From Perfection is a Better Criterion Than Closeness To Evil When Identifying Risky Code. In Proceedings of the IEEE/ACM ASE. ACM, 2010, pp. 113–122.
14. F. Khomh, M. D. Penta, Y.-G. Güeheneuc, G. Antoniol. An exploratory study of the impact of antipatterns on class change- and fault-proneness. Empirical Software Engineering 17(3):243–275, 2012.

15. J. Král, M. Zemlicka. The most important service-oriented antipatterns. In ICSEA, 2007, p. 29.
16. M. Lanza and R. Marinescu, *Object-Oriented Metrics in Practice*. Springer-Verlag, 2006.
17. M. Lorenz and J. Kidd. *Object-Oriented Software Metrics*. Prentice-Hall Object-Oriented Series, Englewood Cliffs, NY, 1994.
18. R. Marinescu. Detecting Design Flaws via Metrics in Object-Oriented Systems. In *Proceedings of TOOLS USA 2001*, pages 103–116. IEEE Computer Society, 2001.
19. R. Marinescu. Detection strategies: Metrics-based rules for detecting design flaws. In *Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04)*. Los Alamitos CA: IEEE Computer Society Press, 2004, pp. 350–359.
20. T.J. McCabe. A Complexity Measure. *IEEE Transactions on Software Engineering*, 2(4):308–320, dec 1976.
21. P. Mihancea. Optimization of Automatic Detection of Design Flaws in Object-Oriented Systems. Diploma Thesis, "Politehnica" University Timisoara, 2003.
22. N. Moha, F. Palma, M. Nayrolles, B. J. Conseil, Y.-G. Guéheneuc, B. Baudry, J.-M. Jézequel. Specification and Detection of SOA Antipatterns. In *International Conference on Service-Oriented Computing (ICSOC)*. Pp. 1–16. 2012.
23. M. Nayrolles; N. Moha; P. Valtchev. Improving SOA Antipatterns Detection in Service Based Systems by Mining Execution Traces in *Proceedings of the 20th Working Conference on Reverse Engineering (WCRE'13)*, pp. 321–330, IEEE, 2013.
24. D. Ratiu, S. Ducasse, T. Girba, and R. Marinescu. Using History Information to Improve Design Flaws Detection. In *Proceedings of the Conference on Software Maintenance and Reengineering (CSMR 2004)*. IEEE Computer Society, 2004.
25. A.J. Riel. *Object-Oriented Design Heuristics*. Addison-Wesley, 1996.
26. A. Rotem-Gal-Oz, *SOA Patterns*, 1st ed. Manning Publications, 2012.
27. A. Trifu. Using Cluster Analysis in the Architecture Recovery of OO Legacy Systems. Diploma Thesis, Karlsruhe and the "Politehnica" University Timis,oara, 2001.

СОДЕРЖАНИЕ

СЕКЦИЯ «АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ»	5
ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ БИЛИНЕЙНОЙ ДИСКРЕТНОЙ СТОХАСТИЧЕСКОЙ СИСТЕМОЙ ПО ВЕРОЯТНОСТНОМУ ИНТЕГРАЛЬНОМУ КРИТЕРИЮ Азанов В.М.	5
ОБ ОЦЕНКЕ ЭФФЕКТИВНОСТИ НЕЙРОСЕТЕВОГО НАСТРОЙЩИКА ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА ПРИ УПРАВЛЕНИИ НАГРЕВАТЕЛЬНЫМ ОБЪЕКТОМ В УСЛОВИЯХ ДЕЙСТВИЯ ВОЗМУЩЕНИЙ Глущенко А.И., Фомин А.В., Еременко Ю.И.	14
РОБАСТНОСТЬ СЛАБО НЕЛИНЕЙНОЙ ДИСКРЕТНОЙ СИСТЕМЫ ПО ОТНОШЕНИЮ К ПАРАМЕТРИЧЕСКИМ ВОЗМУЩЕНИЯМ Даник Ю.Э.	27
КОНЦЕПЦИЯ АРХИТЕКТУРЫ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА ДЛЯ ДИСТАНЦИОННОГО МОНИТОРИНГА АРКТИЧЕСКОЙ ЗОНЫ Емельянова Ю.Г., Хачумов В.М.	39
О ПРИМЕНЕНИИ ВОЗМУЩЁННОЙ ФУНКЦИИ ЛЯПУНОВА К ИССЛЕДОВАНИЮ РОБАСТНОСТИ ДИНАМИЧЕСКИХ СИСТЕМ Каледин М.Л., Макаров Д.А.	54
О РАЗРАБОТКЕ НЕЙРОСЕТЕВОГО НАСТРОЙЩИКА ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА КОНТУРА ТОКА ПРИ УПРАВЛЕНИИ ЭЛЕКТРОПРИВОДОМ ПРОКАТНОЙ КЛЕТИ Петров В.А., Глущенко А.И., Еременко Ю.И.	61
АЛГОРИТМ ИНТЕГРАЦИИ ЗАДАЧ С ГРАФИЧЕСКИМ ТИПОМ ВВОДА И ОЦЕНКА ИХ РЕШЕНИЙ В СИСТЕМАХ УПРАВЛЕНИЯ ОБУЧЕНИЕМ Спиридонов Р.С.	72
МОДЕЛИРОВАНИЕ ПРОЦЕССОВ СЛЕЖЕНИЯ БОРТОВОЙ ВИДЕОКАМЕРОЙ ЛЕТАТЕЛЬНОГО АППАРАТА ЗА ДИНАМИЧЕСКОЙ ЦЕЛЬЮ В ВОЗМУЩЕННОЙ СРЕДЕ Хачумов М.В.	78

СЕКЦИЯ «ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И ПРИНЯТИЕ РЕШЕНИЙ»	85
ПРОБЛЕМЫ И РЕШЕНИЯ РАЗРАБОТКИ СРЕДСТВ АВТОМАТИЗАЦИИ ПОДДЕРЖКИ ДЕЯТЕЛЬНОСТИ ОПЕРАТОРА БУРОВОЙ УСТАНОВКИ Абу-Абед Ф.Н., Борисов Н.А.	85
МЕТОДЫ ИДЕНТИФИКАЦИИ БЛОКИРУЮЩЕГО ПРЕПЯТСТВИЯ В ЗАДАЧАХ ПЛАНИРОВАНИЯ ТРАЕКТОРИИ Андрейчук А.А., Яковлев К.С.	96
РАЗРАБОТКА МОДУЛЯ ЗАДАНИЯ ПРАВИЛ ПРОДУКЦИОННОЙ СИСТЕМЫ Горященко А.С.	105
ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ ОПТИМИЗАЦИИ РАСПОЛОЖЕНИЯ НАРУЖНОЙ ТЕХНОЛОГИЧЕСКОЙ УСТАНОВКИ НА ОБЪЕКТЕ НЕФТЕПЕРЕРАБОТКИ С УЧЕТОМ МИНИМИЗАЦИИ ПОЖАРНЫХ РИСКОВ Гудин С.В., Шихалев Д.В., Хабибулин Р.Ш.	110
СЕТИ ТРАНСФЕРА ТЕХНОЛОГИЙ КАК ЭЛЕМЕНТ ОЦЕНКИ УРОВНЯ ТЕХНОЛОГИЗАЦИИ РЕЗУЛЬТАТОВ ПРИКЛАДНЫХ ИССЛЕДОВАНИЙ Каменская М.А., Тихомиров И.А., Ядринцев В.В., Рутковская И.Б.	121
STRIPS ПОСТАНОВКА ЗАДАЧИ ПЛАНИРОВАНИЯ ПОВЕДЕНИЯ В ЗНАКОВОЙ КАРТИНЕ МИРА Киселёв Г.А., Панов А.И.	131
ДИСКУРСИВНЫЙ АНАЛИЗ В ЗАДАЧАХ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА Ананьева М.И., Кобозева М.В.	139
ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ КАТЕГОРИЙ «РЕСУРС», «СВОЙСТВО», «ДЕЙСТВИЕ», «ОТНОШЕНИЕ» Колесников А.В., Листопад С.В., Денисова А.С., Колонин А.Е.	149
МЕТОД ОБРАБОТКИ ПАРНЫХ СРАВНЕНИЙ ВАРИАНТОВ, ОПИСАННЫХ МУЛЬТИМНОЖЕСТВАМИ Малтугуева Г.С., Петровский А.Б.	161

РАЗРАБОТКА АЛГОРИТМА ОБРАБОТКИ ГРУППОВЫХ МНОГОКРИТЕРИАЛЬНЫХ ЭКСПЕРТНЫХ ОЦЕНОК НА ОСНОВЕ МЕТРИЧЕСКИХ ПРОСТРАНСТВ МУЛЬТИМНОЖЕСТВ	172
Проничкин С.В.	
ГЕТЕРОГЕННАЯ ДИАГНОСТИКА АРТЕРИАЛЬНОЙ ГИПЕРТЕНЗИИ	180
Кириков И.А., Румовская С.Б.	
СЕКЦИЯ «ПАРАЛЛЕЛЬНЫЕ И РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ»	189
СРЕДА КОМПЛЕКСНОГО АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМОВ БАЛАНСИРОВКИ В ПАРАЛЛЕЛЬНОМ МЕТОДЕ ВЕТВЕЙ И ГРАНИЦ	190
Орлов Ю.В.	
ОБЗОР АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ В ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ МЕТОДА ВЕТВЕЙ И ГРАНИЦ	199
Голубева Я.В.	
ПРИМЕНЕНИЕ АППАРАТНОГО УСКОРЕНИЯ НА БАЗЕ ТЕХНОЛОГИИ CUDA В ПРОЕКТАХ ДОБРОВОЛЬНЫХ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ	205
Шутов И.В., Курочкин И.И.	
АНАЛИЗ СЕРВИС-ОРИЕНТИРОВАННЫХ СИСТЕМ: ОБНАРУЖЕНИЕ АНТИПАТТЕРНОВ	208
Югов А.С.	

Научные труды

IV Всероссийская научная
конференция молодых ученых
с международным участием

ИНФОРМАТИКА, УПРАВЛЕНИЕ И СИСТЕМНЫЙ АНАЛИЗ

Труды IV Всероссийской научной конференции
молодых ученых с международным участием

ТОМ I

В авторской редакции

Подписано в печать 05.06.2016. Формат 60×84 1/8. Бумага офсетная.

Усл. печ. л. 27,87. Тираж 100 экз.

Отпечатано в Центре инженерной печати ООО «Наукоемкие технологии»

170023, Тверская область, г. Тверь, ул. Маршала Конева, 12, к. 435.